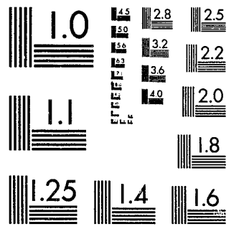

Patrol Car Allocation Model

Program Description

Jan M. Chaiken, Warren E. Walker, Peter Dormont

Rand

The research described in this report was supported by the National Institute of Justice, U.S. Department of Justice under Grant No. 81-IJ-CX-0088.

Published by The Rand Corporation

R-3087/3-NIJ

Patrol Car Allocation Model

Program Description

Jan M. Chaiken, Warren E. Walker, Peter Dormont

July 1985

Prepared for the
National Institute of Justice,
U.S. Department of Justice

Rand
1700 MAIN STREET
P.O. BOX 2138
SANTA MONICA, CA 90406-2138

PREFACE

This report provides installation instructions and an annotated program listing for a computer program written in the FORTRAN language, designed to assist police departments in determining the number of patrol cars to have on duty in each geographical command at different times of the day and week. The program--called the Patrol Car Allocation Model (PCAM85)--is described in two companion reports:

- R-3087/1, *Patrol Car Allocation Model: Executive Summary.*
- R-3087/2, *Patrol Car Allocation Model: User's Manual.*

The first of these, written for police department administrators and planning officials who wish to understand how the Patrol Car Allocation Model can be used in policy analysis, serves as the Summary of both this volume and the User's Manual. The User's Manual provides all the information needed to use the program once it has been installed on a computer system.

The Program Description is written primarily for data processing personnel. Most users will want to read only the first two sections, which describe installation procedures, file formats, memory requirements, and minor program modifications. For the benefit of users who may wish to make substantial modifications, the remainder of the report contains complete documentation, including a program listing with a detailed description of each subprogram. A separate program used to calculate some of the data input for the Patrol Car Allocation Model is also described and listed in App. B; David Jaquette was the coauthor.

Development of the original version of PCAM was supported by the Office of Policy Development and Research of the U.S. Department of Housing and Urban Development (HUD) and by the National Institute of Law Enforcement and Criminal Justice. The modernization was funded by the National Institute of Justice under grant 81-IJ-CX-0088.

CONTENTS

TABLES

FIGURE

GLOSSARY

ALGORITHM
A procedure for performing a calculation.

ALLOCATE
1. Assign a certain number of cars to each shift.

2. Divide a fixed total number of car-hours among shifts.

AMPERSAND (&)
At the end of a line of PCAM instructions, signifies that the command continues on the following line.

ASTERISK (*)
1. At the start of a line of output from the DISP command, indicates that the tour is overlaid by another tour.

2. In input commands, represents the current number of car-hours allocated.

AVAILABLE
1. Ready to be dispatched to a call for service.

2. Not engaged in cfs work or non-cfs work. Not busy.

3. Same as UNCOMMITTED.

BATCH
A mode of operating a computer program in which all instructions are prepared on cards or other input device prior to program execution, and output is received later, usually from a high-speed printer. Contrasted with INTERACTIVE.

BLOCK, TIME
A period of time (whole number of hours) over which the number of patrol cars on duty does not change. One or two time blocks constitute a tour.

BUSY
Unable to be dispatched to a call for service. Busy on cfs or non-cfs work.

CALL RATE
Average number of calls for service received per hour.

CALL RATE PARAMETER
A parameter for each day in each precinct. When multiplied by the hourly call-rate factor, gives the expected number of calls for service in the hour.

CAR (see PATROL CAR)

CAR-HOUR
One patrol car on duty for one hour.

CFS
Call(s) for service.

CFS WORK
1. All activities of a patrol car from the time it is dispatched to a call for service until the time it is available again for dispatch.

2. Number of car-hours spent on such activities.

CFS WORKLOAD
1. Loosely speaking, the extent to which cfs work is a burden on a patrol car.

2. Technically, the number of car-hours of cfs work in a given period of time.

COMMAND
1. An instruction to the PCAM program.

2. An administrative unit in a police department that is supervised by a superior officer. (Used in the expression *geographical command*.)

CONSTRAINT
A number specified as the largest or smallest value permitted for a performance measure.

CURRENT-DATA
Some or all of the data in DATABASE, which have been read into the computer memory by a READ command and are used and/or modified by PCAM commands.

DATABASE
The data prepared by the user for input into PCAM.

DAY
A 24-hour period used for organizing PCAM data. Not necessarily a calendar day; in fact, a DAY should usually be started at a time when there are few calls for service, for example, 0400, 0500, 0600, 0700, or 0800.

DELAY, TOTAL
Sum of queuing delay and travel time. (Same as TOTAL RESPONSE TIME.) Shown in headings as "QUEUE +TRVL".

DELIMITER
Any character other than a letter, digit, parenthesis, asterisk, hyphen, period, or ampersand. Examples of delimiters are blanks, commas, colons, and equal signs.

DESCRIPTIVE MODE
Capability of the PCAM program to calculate and display performance measures by time of day and geographical command when the numbers of patrol cars on duty in each shift have been specified.

DIVISION
A combination of precincts. Some police departments use the word "division" for a precinct. This is permitted in PCAM by changing the keyword PRECINCT.

EFFECTIVE CAR
The equivalent of a patrol car that does not engage in any non-cfs work.

EXPONENTIALLY DISTRIBUTED
A random variable T is exponentially distributed if there is a parameter μ such that

$$\text{Prob}(T > t) = e^{-\mu t}.$$

The mean of T is $1/\mu$. The assumption that service times for calls to the police are exponentially distributed is not verified by data, but the assumption is technically necessary in PCAM. (This is a source of PCAM's simplicity.)

FIELDED
In the field. A patrol car is fielded if it is on duty.

FILLER WORD
One of the following words, which may be entered in a PCAM command if desired, but will be ignored by the program: FOR, CAR, HOUR, HOURS, TO, ON, BY, DATA.

HOURLY CALL RATE FACTOR
A parameter for a single hour in a single precinct. When multiplied by the call rate parameter for the day, gives the expected number of calls in the hour.

HOURLY SERVICE TIME FACTOR
A parameter for a single hour in a single precinct. When multiplied by the service time parameter for the day, gives the expected service time (in minutes) for calls received during the hour.

INTERACTIVE
A mode of operating a computer program whereby the user enters instructions at a terminal and receives output immediately at the same terminal. Contrasted with BATCH.

KEYWORD
A character string that has a special meaning to the PCAM program. These are either filler words or one of the following: DAY, P, C, T, F, ADD, ALOC, DISP, END, HEADR, LIST, MEET, READ, SET, WRITE, TOUR (or a substitute provided by the user), DIVISION (or a substitute), PRECINCT (or a substitute).

LIMITING CONSTRAINTS
When meeting constraints, the performance measures whose constrained values lead to a need for the largest number of patrol cars. (If these constraints were eliminated, a smaller number of patrol cars would meet all the constraints.)

LIST
Command that causes PCAM to print out the values of the data items associated with all precincts, days, and tours within its scope.

MINIMUM ALLOCATION
The smallest whole number of actual patrol cars that can be assigned to a shift to handle the call-for-service workload.

NEW-DATA
A permanent file created by the WRITE command from all or part of CURRENT-DATA.

NON-CFS WORK
1. Any activity of a patrol car that makes the car unavailable for dispatch but was not generated by a previous dispatch to a call for service.

2. Number of car-hours spent on such activities.

OBJECTIVE FUNCTION
The performance measure to be minimized by an allocation.

OFFICER HOUR
One police officer on duty for one hour.

OPTIMAL
Yielding the smallest possible value of the objective function.

OUTPUT ORDER
A choice of displaying output tables either by tour within day within precinct, or by precinct within tour within day.

OVERLAY TOUR
A tour that begins during one tour and ends during the following tour.

PARAMETER
A number that characterizes a particular hour, block, shift, day, or precinct. See also SERVICE TIME PARAMETER and CALL RATE PARAMETER.

PATROL CAR
A mobile vehicle that can respond to calls for service from the public. Includes vehicles other than automobiles that serve the same function, e.g. scooters.

PATROL INTERVAL
The interval (hours:minutes) between successive times that a random point will be passed by a car, if all uncommitted time is devoted to random preventive patrol.

PCAM
Patrol Car Allocation Model.

PLUS (+)
1. At the start of a line of output from the DISP command, indicates that the tour is an overlay.

2. In the heading +TRVL, means that travel time is added to queuing delay.

POISSON PROCESS
In the PCAM context, the occurrence of calls for service in a given precinct during a given hour constitutes a Poisson process if there is a parameter λ such that the time between calls has the distribution

$$\text{Prob}(\text{time between calls} > t) = e^{-\lambda t}.$$

This assumption is well verified by data.

PRECINCT
A geographical area that is treated as independent from other areas by the patrol car dispatcher. Each patrol car is assigned to an entire tour in one precinct, although it may work in only part of the precinct.

PRESCRIPTIVE MODE
Capability to suggest the number of patrol cars that should be on duty during each shift, so as to meet standards of performance specified by the user.

PREVENTIVE PATROL
The practice of driving a patrol car through an area, with no particular destination in mind, looking for criminal incidents or opportunities, suspicious occurrences, etc.

PRIORITY
Importance of a call for service. PCAM permits three priority levels. Priority 1 calls are so important that the dispatcher will violate ordinary dispatching practices to get a patrol car to respond immediately. The PCAM program ignores these special efforts of dispatchers and may, as a result, indicate delays that are somewhat higher than actual for priority calls. Priority 2 calls are important enough that a rapid response is preferred over a slow response. Priority 3 calls can wait in queue without deleterious effect.

QUALIFIER
Phrase(s) associated with a computer command, defining the scope of the command. May be any subset of these phrases, separated by delimiters: 'TOUR=<NAMELIST>', 'DAY=<NAMELIST>', 'DIVISION=<NAMELIST>', 'PRECINCT=<NAMELIST>'.

QUEUE
In the PCAM context, a collection of calls for service that are waiting to be assigned to a patrol car because no patrol car is available at the moment.

QUEUING DELAY
The length of time a call for service waits in queue.

REGRESSION ANALYSIS
A procedure for fitting a straight line to data so as to minimize the sum of the squares of the deviations of the data from the straight-line estimate.

RESPONSE TIME, TOTAL
Sum of queuing delay and travel time. (Same as TOTAL DELAY.)

SCOPE
The collection of precincts, tours, and days to which the action of a PCAM command applies.

SERVICE TIME
Number of minutes a patrol car will be unavailable from the time it is dispatched to a call until it is available to respond to another call.

SERVICE TIME PARAMETER
A parameter for each day in each precinct. When multiplied by the hourly service time factors, gives the expected service time in each hour.

SHIFT
A particular tour in a particular precinct on a particular day.

SMOOTHING
A method of calculating delays, available in PCAM by user option. Queuing behavior is smoothed over time by averaging queuing delays and queuing probabilities in two successive hours of a day. If smoothing is not chosen, the PCAM program calculates delays as if each hour was in steady state.

SQUARE-ROOT LAW
An equation for the average travel distance D in a region of area A when N patrol units are available:

$$D = (\text{constant}) \times \sqrt{\frac{A}{N}}.$$

STEADY STATE
In the PCAM context, a situation where the probability of finding n cars available does not change over time.

TIME BLOCK
See BLOCK, TIME

TOTAL DELAY
Same as RESPONSE TIME, TOTAL; the sum of queuing delay plus travel time.

TOUR
A period of time (whole number of hours) beginning when a patrol officer starts work for the day and ending when the officer finishes work. In PCAM, tours are assumed to start at the same time in every precinct on every day (but overlay tours need not be present on every day in every precinct).

TRAVEL TIME
The length of time from the moment a patrol car is dispatched to an incident until the moment it arrives at the scene.

UNAVAILABILITY PARAMETERS
A pair of constants B1 and B2 for each precinct that give the best regression fit to the linear equation

$$\begin{pmatrix}\text{fraction of time}\\ \text{on non-cfs work}\end{pmatrix} = B1 \times \begin{pmatrix}\text{fraction of time}\\ \text{on cfs work}\end{pmatrix} + B2$$

UNCOMMITTED TIME
The minutes or hours during a tour when a patrol car is not engaged in either cfs work or non-cfs work. This time can be used for directed patrol, preventive patrol, or any activity that does not make the car unavailable for dispatch.

UTILIZATION
The fraction of time a patrol car is busy on cfs work.

I. PROGRAM INSTALLATION

INTRODUCTION

The Patrol Car Allocation Model (PCAM85) is a computer program designed to help police departments determine the number of patrol cars to have on duty in each of their geographical commands. Typically, the number of patrol cars needed will vary according to the season of the year, day of the week, and hour of the day.

A companion User's Manual describes applications of the program, explains the meaning of the various items of data to be included in the database, and gives complete instructions for operating the program once it is installed:

- Jan M. Chaiken and Warren E. Walker, *Patrol Car Allocation Model: User's Manual,* R-3087/2.

The PCAM program is written in the FORTRAN language and is compatible with most FORTRAN compilers.

Successful use of the PCAM program requires little or no expertise in the use of computers. The user controls the program with a sequence of simple commands. These can be read in and stored for operation in batch mode (where the program's output is produced on a line printer), or they can be entered one at a time at a terminal for operation in interactive mode (in which case the program's output is displayed immediately at the terminal). Some of the facilities provided by the commands are:

- Data selection
- Allocation of patrol cars to meet constraints on performance measures
- Allocation of patrol cars to best achieve specified objectives
- Display of measures describing expected patrol car performance under particular allocations.

The data required for processing these commands must be supplied to the program in an external file that we call DATABASE. The format for this file is described in Sec. II.

Installing PCAM on a computer system is a simple and straightforward operation. However, various computer systems differ with respect to their conventions for accessing files in the FORTRAN language, and this may have to be taken into account in the program installation. In addition, users may wish to optimize the amount of run-time storage reserved, with respect to the size of their database and intended use of the program.

The program, as listed in Sec. V and distributed by Rand, is set up to run in batch mode, with changes for interactive mode indicated (see "Minor Program Modifications," below). However, on request we will supply the program in a form suitable for interactive operation. If the program is to be used primarily in batch mode, many users will wish to make program changes to enhance the appearance of the output.

This section provides the information needed to install the program and make the indicated types of changes. The user wishing to make more substantial changes will have to familiarize himself with Secs. IV and V. Refer to the Glossary and the User's Manual for definitions of unfamiliar terms.

PCAM SOURCE LANGUAGE AND COMPILATION

The PCAM program is written in the FORTRAN language. The program conforms closely to ANSI[1] standards, and the previous version (PCAM75) operated successfully on many different makes of computer hardware.

When all desired modifications to the source code have been made, the PCAM program should be compiled and the object program saved in an execution-ready form. On IBM 360, 370, or 303X systems running under OS or similar operating systems, the following JCL might be used to accomplish this:

[1]American National Standards Institute (ANSI), *FORTRANfR, X-3.9-1966 (also known as FORTRAN*66) and ANSI *FORTRAN*, X-3.9-1978 (also known as *FORTRAN*77).


```
//      jobcard
//STEP1 EXEC FORTGCL
//FORT.SYSIN DD *
            .
            .
            .
          PCAM source program
            .
            .
            .
/*
//LKED.SYSLMOD DD    definition of load module library
//LKED.SYSIN DD *
    NAME PCAM
/*
```

FILE STRUCTURE AND CONVENTIONS

The basic inputs to the PCAM program are (1) a sequence of commands, supplied by the user from a stored file or through an interactive terminal, which control the functions performed by the program, and (2) the DATABASE file on a direct access or magnetic tape device, which describes the characteristics of a city that are relevant to PCAM's modeling of its police patrol operations. PCAM's basic operations can be performed only on the part of DATABASE that resides in the computer's main memory. The user directs part or all of the data to be read from DATABASE by means of a READ command, as described in the User's Manual. The term CURRENT-DATA refers to the data that have been read from DATABASE and are available for processing.

The user can modify the contents of CURRENT-DATA through various commands. PCAM also has the capability of writing out a file containing part or all of the information in CURRENT-DATA. The file created by this operation is called NEW-DATA and is written in response to a WRITE command (see User's Manual). It is in the same format as DATABASE and can be used in its place in subsequent runs of PCAM.

PCAM references all files through INTEGER variables that contain FORTRAN unit numbers. This facilitates changing the unit numbers used by PCAM to conform to the conventions of a particular operating system.

Table 1 describes PCAM's files in terms of these file reference variables and gives the values of the variables in the distributed program. All file reference variables (except the variable for NEW-DATA) are in COMMON/SYSTEM/. To change the value of these file reference variables, the DATA statement numbered 5374 in Sec. V, located in the BLOCK DATA subprogram, should be modified.

Whether the user changes the unit numbers or not, most operating systems require the user to prepare data definition statements that identify the device or file corresponding to each unit number. For example, the following JCL is used to run the program on the Rand Computation Center's IBM 3032 computer:

Table 1

PCAM FILES

Variable Name	Device	Description	Unit Number
SYSIN	Teletype, terminal, card reader, disk, tape, etc.	User's command input	10
SYSOUT	Teletype, terminal, printer, disk, tape, etc.	Printed output from program	11
IFILE	Tape, direct access	Input data (DATABASE)	19
NUNIT	Tape, direct access	Output database (NEW-DATA)	Value determined from user input in the WRITE command
LIT	Tape, direct access	Scratch file for literals (space needed for three 80-character records)	20


```
//     jobcard
//S1   EXEC PGM=PCAM,REGION=160K
//STEPLIB DD DSN= name of load module library,DISP=SHR
//GO.FT10F001 DD DSN=name of command file,UNIT=USER,DISP=SHR
//GO.FT11F001 DD SYSOUT=A,DCB=(RECFM=FA,LRECL=81,BLKSIZE=81)
//GO.FT06F001 DD SYSOUT=A
//GO.FT18F001 DD DSN=name of NEW-DATA file,UNIT=USER
//    VOL=SER=volume,SPACE=(TRK,(4,1),RLSE),DISP=(NEW,CATLG),
//    DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//GO.FT19F001 DD DSN=name of DATABASE file,UNIT=USER,DISP=SHR
//GO.FT20F001 DD DSN=&&PCAM,UNIT=TEMP,VOL=SER=TEMP10,
//    SPACE=(TRK,(10,2),RLSE),DISP=(NEW,DELETE)
```

The FT18 DD statement allows the user to write a NEW-DATA file on FORTRAN unit 18. In other words, the user is permitted to enter the command WRITE [DATA] [ON] 18 [FOR] <QUALIFIER>.

STORAGE ALLOCATION

Most run-time storage that PCAM uses is allocated dynamically from two large one-dimensional arrays (see Sec. IV). The arrays are named CDAT (an abbreviation for CURRENT-DATA) and C2DAT, and are contained in COMMON/STORE/. Wherever COMMON/STORE/ occurs in the program, arrays ICDAT and IC2DAT of the same size as CDAT and C2DAT are defined and equivalenced to CDAT and C2DAT. The minimum amount of storage that must be reserved for CDAT and C2DAT depends on the size of the user's database and on how much of the database will be accessed in a single READ command.

Four different classes of information are stored in CDAT. The storage requirements for each class are given below. The sum of these requirements is the minimum size for array CDAT. C2DAT requires the same amount of space as CDAT.

1. Permanent Tables

Tables that are allocated at the start of each PCAM run require the following number of words of storage:

$$(9 \cdot NDAYDT) + (13 \cdot NTRDT) + (3 \cdot NBLDT) + (8 \cdot NDIVDT),$$

where NDAYDT = number of days of data in DATABASE, NTRDT = number of tours in each day in DATABASE, NBLDT = number of blocks for each day in DATABASE, and NDIVDT = number of divisions in DATABASE.

2. Variable Size Tables

Tables whose size depends on the number of divisions, days, and tours selected in a READ command qualifier require the following number of words of storage:

$$NDAYRD + NTRRD + NDIVRD,$$

where NDAYRD = number of days read into CURRENT-DATA, NTRRD = number of tours read, and NDIVRD = number of divisions read.

3. Data Storage

Data read from DATABASE into CURRENT-DATA by a READ command requires the following number of words of storage:

$$NPCTRD \cdot NWDPCT,$$

where NPCTRD = number of precincts included in CURRENT-DATA, and the calculation of NWDPCT will be explained below in Sec. IV (Table 7).

4. Temporary Storage

Temporary storage is used for names and numbers[2] during command interpretation and execution. The exact amount of this storage that is needed for any command is given by:

$$8 \cdot (\text{number of names in command}) + 2 \cdot (\text{number of numbers in command}).$$

[2]Names appear in user commands to label entities such as precincts, days, and tours. Numbers are used to identify output tables, objective functions, constraints, files, and numerical quantities. For more detailed information, see Sec. II and also the User's Manual.

Temporary storage is always released when the execution of a command is complete. An allocation of about 150 words for this type of storage will be sufficient for most applications.

The program as distributed allows 6,000 words for the sum of these four requirements. Most users will find this amount of space adequate. If the user's DATABASE is too large, an error message will be printed when the program is run, and then it will be necessary to calculate the actual requirements as listed above.

If the user's DATABASE requires less than 6,000 words for CDAT, the program will operate properly, and a message will be printed after the END command indicating the total number of words actually used for the first three requirements listed above. The user can then reduce the amount of space allocated to CDAT and C2DAT, if desired. The only advantage in making this modification will be a possible reduction in the cost of running the program, if the computer installation's charges depend on the amount of memory requested.

In order to change the space allocation, either an increase when necessary or a decrease when desired, the dimensions of CDAT, ICDAT, C2DAT, and IC2DAT must be changed on the following pairs of lines of the program (see Sec. V):

8, 10	2886, 2888
272, 274	2928, 2930
407, 409	2978, 2980
490, 492	3045, 3047
563, 565	3252, 3254
598, 600	3439, 3441
1144, 1146	3844, 3846
1404, 1406	3900, 3902
1544, 1546	3950, 3952
1695, 1697	3992, 3994
1794, 1796	4125, 4127
1966, 1968	4288, 4290
2085, 2087	4386, 4388
2235, 2237	4534, 4536
2341, 2343	4571, 4573
2520, 2522	5085, 5087
2578, 2580	5117, 5119
2800, 2802	5365, 5367
2840, 2842	

In addition, in BLOCK DATA, NWORDS must be set equal to the dimension of CDAT. This occurs on line 5370.

MEMORY REQUIREMENTS

A moderate amount of computer memory is required to run the PCAM program. Although the exact amount will vary from one computer system to another, the figures given below will serve as a good guideline.

The memory requirements for the PCAM program depend directly on the size of the array CDAT; call this NWORDS. The amount of storage required to run PCAM on the Rand Computation Center IBM 3032 computer is given by: (193 + 8 • NWORDS/1024) K bytes. The program as distributed has NWORDS = 6,000, and therefore requires 240K bytes of storage, but we suggest requesting 242K bytes. For installations with other types of computers, the equivalent requirement can be obtained by using the fact that there are four bytes in a word on the IBM 3032 computer and that 1K byte = 1024 bytes.

For the assistance of potential users who are severely restricted in the amount of storage, we point out that the memory requirements for PCAM can be reduced by means of a technique called chaining.[3] (We do not recommend chaining the PCAM program unless it is unavoidable.) This technique allows only those parts of the program that are required to perform a particular function to be resident in memory. The remainder of the program can remain in external storage until required. For example, an examination of the program listing in Sec. V and the cross-referenced listing of program segments in App. D reveals that while a LIST command is being executed (by subroutine LIST) there is no need for subroutine WRITE, which implements the WRITE command, to be resident in memory.

The best way to break up the PCAM program for chaining will vary from installation to installation and depends upon the amount of memory available to run the program and the way in which it is used. In

[3]This is frequently called "overlaying," but we do not use this terminology because the word "overlay" has been assigned a specific (and different) meaning in the User's Manual and the Glossary.

general, those subprograms required for the execution of all, or most, commands should be in memory throughout the program's execution. The subprograms that are needed to execute particular commands should be grouped so that only the group required to execute one command resides in memory with the continuously resident subprograms.

MINOR PROGRAM MODIFICATIONS

To convert the PCAM batch program into an interactive program, five lines must be removed from the program (1859, 1860, 2551, 2552, and 2561), and two lines must be inserted (2533 and 2534). These are described in Sec. V under the headings "MAIN Program" and "Subroutine GETTKN."

If the program is operated in batch mode, the user may wish to change the appearance of output, because there is more room on a page of output from a high-speed printer than there generally is on the screen of an interactive terminal. Changes in spacing of columns of output, number of decimal places displayed, and column labels may be made as follows:

- For the table displayed by the LIST command, change format statements in subroutine LIST.
- For the tables displayed by the DISP command, change format statements in subroutine TITLE and subroutine PRTBL (print table).

In addition, output data of no interest to the user can be completely suppressed by modifying the same subroutines.

To change pagination, or to provide a different heading at the top of the first page (for example, the name of the police department or the date of the run), modify the MAIN program or the INIT subroutine.

If all tours[4] have the same length, the department may prefer to have PCAM allocate *cars* rather than *car-hours*. The program as

[4]In the PCAM documentation, the word "tour" is used to designate the period of time during which a patrol car is on duty. The program itself employs whatever term the user specifies--watch, shift, platoon, etc.

distributed will accept commands referring to cars, such as ALOC 24 CARS BY F(2), but it will interpret the expression "24 cars" to mean 24 car-hours. To change the program so that the input number is interpreted as cars, line 141 of the program must be changed following the instructions on lines 136-140 (see Sec. V).

COSTS

The cost of running the PCAM program will vary from installation to installation. However, we can give a rough idea of the range of costs based on our experience with two computer systems. On the Rand Computation Center IBM 3032 computer, compiling the program costs approximately $14, and this is more expensive than most runs of the program after compilation. (It is therefore desirable to save the object code from the compiled program.)

The demonstration of the program illustrated in the figures in Sec. III of the User's Manual was run from object code at a cost under $12. Typical applications should involve costs under this amount for machine time unless numerous ADD or ALOC commands are entered, or these commands are performed on a large number of shifts simultaneously. (In PCAM terminology, a *shift* is a tour in all precincts at once. The number of shifts is the product of the number of precincts, days, and tours included explicitly or implicitly in a command qualifier.) Use of PCAM's "smoothing" option (see App. A of the User's Manual) will double the cost of a run. In general, PCAM is an inexpensive program to operate and compares favorably with any other program that could answer similar policy questions.

II. PCAM DATA FILE FORMAT

This section describes the format of the DATABASE and NEW-DATA files mentioned in Sec. I. Figure 1 and the demonstration DATABASE in App. A may assist the user in interpreting the instructions in this section. The format items shown are those used to read the DATABASE file. Those used to write NEW-DATA files are different in some respects as noted, but they always produce a file that can later be read according to the formats shown for DATABASE. The reader is referred to the Glossary and the User's Manual for the definitions of unfamiliar terms. Appendix B documents a computer program (not part of PCAM) that may assist some departments in calculating values for the unavailability parameters B1 and B2 that appear in the precinct header record, described below.

The DATABASE file must be prepared in standard 80-column records on a disk or other rewindable storage device. The PCAM program uses the variable name IFILE for DATABASE and assumes it is located on unit 19. The user may change the unit number in COMMON/SYSTEM/, which is initialized on line 5374 in BLOCK DATA (see Sec. V).

1. *Control record.* This is the first record in the database.

Columns	*Format*[1]	*Comments*	*Description*
1-8	A8	Left justify	The word DIVISION, or whatever word the department uses for aggregations of precincts.
11-18	A8	Left justify	The word PRECINCT, or whatever word the department uses for precincts.
21-28	A8	Left justify	The word TOUR, or whatever word the department uses for tour.
30-31	I2	Right justify	Number of divisions in the database.
33-35	I3	Right justify	Number of precincts in the database.

[1]All A8 formats are read as 8A1.

37-39	I3	Right justify	Number of days of data that are supplied for each precinct.
41-42	I2	Right justify	Number of time blocks in each day.
44-45	I2	Right justify	Number of tours in each day.
47	I1		Indicator for overlay tour. Enter 0 or 1 as described below.

PCAM permits the following possibilities for overlay tours:

TYPE 0 -- there are no overlay tours.
TYPE 1 -- (a) every day in every precinct has a single overlay tour
(b) some days and/or precincts have a single overlay tour; the remainder have none.

Enter 0 as the overlay tour indicator for Type 0; enter 1 for Type 1. For Type 1, the last tour in the data for every day in every precinct must be the overlay tour. However, in case 1(b), the overlay tour data will be blank for some days and/or precincts.

Columns	*Format*	*Comments*	*Description*
49	I1		Error checking flag. Enter a 1 for no error checking. If blank or 0, error checking of the input data will be performed, as discussed in the description of subroutine READ.
51	I1		Smoothing flag. Enter a 1 if smoothing of queuing behavior is desired. If blank or 0, no smoothing will be performed. (See App. A of the User's Manual for a description of smoothing.)
53	I1		Indicator for the presence or absence of a "filename" record in the database. Enter a 1 if the second record in the database is a "filename." Blank or 0 means no filename is being supplied.

2. *Filename record*. This is the second record in the database if there is a "1" in column 53 of the Control record. The filename can consist of up to 60 characters, which identify the database being used in a particular run. If supplied in this record, the filename will be printed at the top of every PCAM output report.

Columns	*Format*	*Comments*	*Description*
1-60	60A1		Name of database.

3. *Day name record(s)*. If there are ten days or fewer in the database, then only one day name record is required. Otherwise, continuation records will be needed; supply as many as required.

Columns	*Format*	*Comments*	*Description*
Begin in 1	10A8	Left justify	Name for each day in the database. For each precinct, day data will have to be in the same order as the names on this record.

4. *Block descriptor record*. This follows the day name record(s).

Columns	*Format*	*Comments*	*Description*
Begin in 1	24(I2,1X)	Right justify	Last hour of each time block. Supply as many hours as there are time blocks in a day, up to 24. The hours must be in increasing order.

5. *Tour descriptor records*. There is one such record for each tour. These records follow the Block Descriptor Record. The record for the overlay tour (if any) is last.

Columns	*Format*	*Comments*	*Description*
1-8	A8	Left justify	Name of tour.

10-11	I2	Right justify	Ordinal number of the first time block (or only time block) in the tour.
13-14	I2	Right justify	Ordinal number of the second time block in the tour. Zero or blank if the tour has only one time block.

6. *Precinct header record.* The first such record follows the last tour descriptor record. The next such record follows all the data for the first precinct. In total, the number of precinct header records will equal the number of precincts in the data.

Columns	*Format*	*Comments*	*Description*
1-8	A8	Left justify	Precinct name.
10-17	A8	Left justify	Division name for this precinct.
20-24	F5.0[2]		Area of precinct (in square miles).
26-30	F5.0[3]		Total length of streets in precinct (in miles).
32-36	F5.0[4]		Unavailability parameter B1.
38-42	F5.0[4]		Unavailability parameter B2.

7. *Day detail records.* There are three of these records for each day in each precinct. The first three follow the first precinct header, the next three appear after all data for shifts and blocks in the first day in the first precinct, etc.

Record	*Columns*	*Format*	*Description*
1	1-5	F5.0	Call rate parameter.
	7-11	F5.0	Service time parameter.
	13	I1	An indicator for the presence or absence of an overlay tour for this day for this precinct. Enter 0 if there is no overlay tour, 1 if there is an overlay tour.
2	1-72	24(F3.2)[5]	Call rate factors for each hour of the day. The product of one of these factors and the call rate parameter in record 1 should be the number of calls occurring in the precinct in the corresponding hour of the day.
3	1-72	24(F3.2)[5]	Service time factors for each hour of the day. The product of one of these factors and the service time parameter in record 1 should be the average service time for calls occurring in the precinct in the corresponding hour of the day.

[2] F5.2 in NEW-DATA.
[3] F5.1 in NEW-DATA.
[4] F5.3 in NEW-DATA.

8. *Shift detail records.* There is one such record for each tour for each day for each precinct. After the third day detail record for each day in each precinct, there will be N of these records, describing the N tours in that day.

Columns	*Format*	*Description*
1-5	F5.0[6]	Average number of cars on duty during the shift.
7-11	F5.0[6]	Average speed of cars when responding to calls (in miles/hour).
13-17	F5.0[6]	Average speed of cars when on preventive patrol (in miles/hour).
19-23	F5.0[7]	Fraction of calls that are of priority 1.
25-29	F5.0[7]	Fraction of calls that are of priority 2.

[5] No decimal points in NEW-DATA.
[6] F5.1 in NEW-DATA.
[7] F5.3 in NEW-DATA.

31-35	F5.0[8]	Percent of cars with two officers. The remainder of the cars will be assumed to have one officer. (If this field is blank, it will be assumed that all cars have one officer.)
37-41	F5.0[9]	Fraction of priority 1 calls to which two cars are dispatched (see below).
43-47	F5.0[9]	Fraction of priority 1 calls to which three cars are dispatched (see below).
49-53	F5.0[9]	Fraction of priority 2 calls to which two cars are dispatched (see below).
55-59	F5.0[9]	Fraction of priority 2 calls to which three cars are dispatched (see below).
61-65	F5.0[9]	Fraction of priority 3 calls to which two cars are dispatched (see below).
67-71	F5.0[9]	Fraction of priority 3 calls to which three cars are dispatched (see below).

PCAM85 allows different numbers of cars to be sent to calls of different priorities, and to different types of calls within each priority. Columns 37-71 of a shift detail record allow the user to specify the dispatch policy for a shift. There are two input fields for each priority: the fraction of calls that are dispatched two cars and the fraction that receive three cars. (PCAM assumes that the remainder of the calls receive one car.) If the two fields for any priority are left blank, PCAM assumes that a single patrol car is dispatched to all calls of that priority.[10]

9. *Blank records*. There is one blank record for each day for each precinct. It must follow the shift detail records for that day and precinct.

---

[8]F5.1 in NEW-DATA.

[9]F5.3 in NEW-DATA.

[10]This is PCAM's simplified way of representing important aspects of dispatching policies that are, in most police departments, much more complex. In designing PCAM, we considered other ways of representing dispatch policies in the computer program. We found the more precise ones were difficult to describe with input data, and the resulting output reports were difficult to understand.

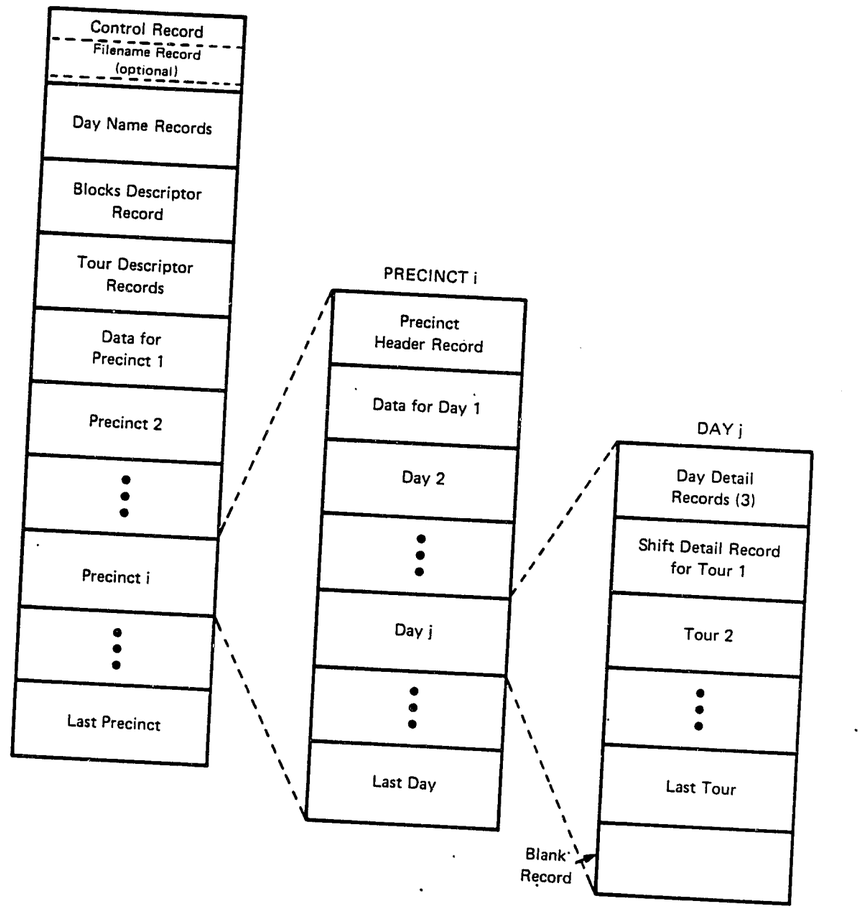

Fig. 1 — Order of data in DATABASE

III. ALGORITHMS FOR CONVERSION OF ALLOCATION BETWEEN TOURS AND BLOCKS

This section documents the algorithms that PCAM uses to convert allocations of cars to tours into block allocations, and vice versa. Recall that a tour is a period of time over which a patrol car can be on duty, and a block is a part of a tour during which the number of patrol cars on duty is constant.

CONVERSION OF TOUR ALLOCATIONS TO BLOCK ALLOCATIONS

Given an allocation of cars to the tours of a day in a precinct, PCAM determines the resulting allocation of cars to blocks as follows:

1. Set the number of cars assigned to each block of the day to zero.
2. For each tour of the day, including overlay tours if any, add the number of cars assigned the tour to the number of cars assigned to each of its blocks. For example, consider four blocks named 1, 2, 3, and 4, and three tours named A, B, and C. Tour A works blocks 1 and 2, tour B works blocks 2 and 3, and tour C works blocks 3 and 4. Let N_i be the number of cars on duty during block i, and let M_A be the number of cars assigned to tour A. Then $N_1 = M_A$, $N_2 = M_A + M_B$, $N_3 = M_B + M_C$, $N_4 = M_C$.

CONVERSION OF BLOCK ALLOCATIONS TO TOUR ALLOCATIONS

PCAM uses two different algorithms to convert block allocations to tour allocations. One algorithm is used to determine the allocation of cars to tours after constraints have been met for blocks. The second algorithm is used to determine the required increase in car allocation when the number of cars allocated to tours is not enough cars to handle the call-for-service workload in all blocks of a day.

When Constraints for Blocks Are Met

The algorithm to determine the tour allocation after constraints are met for all blocks of a day in a precinct is the following. For each tour of the day:

a. If the tour is not involved in an overlay, assign the maximum of the number of cars in its blocks.
b. If an overlay tour starts during the tour, assign the number of cars assigned to its first block. Save this as N_1. Save the number of cars assigned to its second block as N_2.
c. If the overlay tour ends during the tour, assign the number of cars assigned to its second block. Save this as N_4. Save the number of cars assigned to its first block as N_3.
d. If the tour is an overlay tour, assign $N = \max(N_2 - N_1, N_3 - N_4, 0)$ cars.
e. If N equals 0, STOP.
f. Let $\delta = \max(N_2 - N_1, 0) - \max(N_3 - N_4, 0)$.
g. If $\delta > 0$ and the overlay tour is longer than the first overlaid tour, add δ cars to the first overlaid tour and remove δ cars from the overlay tour. If $\delta < 0$ and the overlay tour is longer than the second overlaid tour, add δ cars to the second overlaid tour and remove δ cars from the overlay tour. If $\delta = 0$ or the overlay tour is not longer than the overlaid tour, make no adjustments.

When Block Allocation Is Insufficient To Handle Workload

Under conditions where the allocation of cars to the blocks of a day is insufficient to handle the call-for-service workload in each block (this can result from the execution of a READ or SET command), the following algorithm is used to determine where to increase the assignment of cars.

a. Determine the minimum number of cars required for each deficient block and assign that number of cars. Mark such blocks as deficient.

b. For each tour of the day (the overlay tour last):

   i. If the tour is not involved in an overlay and has a deficient block, assign the maximum number of cars assigned to its blocks.

   ii. If an overlay tour starts during the tour, save the number of cars assigned to its first block as N_1. If its first block is deficient, assign N_1 cars to the tour.

   iii. If an overlay tour ends during the tour, save the number of cars assigned to its second block as N_4. If its second block is deficient, assign N_4 cars to the tour.

   iv. If the tour is an overlay and one or both of its blocks is deficient, assign cars as follows. Let N_2 be the number of cars assigned to its first block and N_3 be the number of cars assigned to its second block (these include the effect of increasing a block assignment to meet the workload restriction in step a. above. However, note that at this stage in the algorithm, any assignments made to tours in steps b. and c. have not been converted to block assignments). The number of cars assigned to the overlay tour is then $\max(N_2 - N_1, N_3 - N_4,$ number cars currently assigned).

c. The algorithm described above (Conversion of Tour Allocations to Block Allocations) is then used to redetermine the block assignments.

IV. INTERNAL DATA STRUCTURES

This section describes PCAM's internal data structures and its run-time storage management system. An understanding of these aspects of the program is necessary only if the user wishes to interpret the program listings in Sec. V or modify the program. This section assumes a familiarity with the database format given in Sec. II. Refer to the Glossary and the User's Manual for definitions of unfamiliar terms.

STORAGE MANAGEMENT

The amount of memory required by PCAM will vary according to the size of the database and the portion of it selected in each READ command. To allow for this variation while enabling the program to run with the minimum amount of storage required for a particular database, PCAM dynamically allocates much of the storage that it uses.

Dynamic storage allocation is accomplished by reserving two large, one-dimensional arrays, the size of which can be set when the program is compiled. Then, when a variable amount of storage is required for some purpose, it can be allocated from these arrays, at which time the subscript of the first word allocated is saved for future reference. The arrays are referenced by the variable names CDAT and C2DAT when REAL data are accessed and by ICDAT and IC2DAT when INTEGER data are accessed.

Two subroutines are used to allocate storage from CDAT; these are GETBOT and GETTOP. GETBOT allocates storage from the "bottom" of the array. This storage is used for three types of data: (1) tables whose dimensions depend only on certain parameters describing the database and do not vary during program execution, (2) tables whose dimensions can change as a result of the number of days and tours selected in a READ command, and (3) data read from the database by a READ command. Subroutine GETTOP allocates storage from the "top" of CDAT. This is basically "scratch pad" storage, used during interpretation of all commands and sometimes during command execution.

The storage management system is actually rather simple. Storage is allocated on a last-in-first-out basis. Each routine that requests storage has the responsibility of releasing it or not, depending upon intended future use. Storage is released by setting a pointer to a subscript that represents the highest or lowest free word of CDAT, depending upon whether storage is being freed from the top or bottom. Thus, care has been exercised so that storage is not prematurely freed and unrecoverable "holes" are not left in allocated storage.

TABLE POINTERS

Pointers to the dynamically allocated tables used by PCAM and table dimensions are saved in COMMON/PNTRS/. This common block also contains certain variables relating to overlay tours. For completeness, these variables will also be described here. Table 2 lists each variable in COMMON/PNTRS/, its contents, and the routine where its value is set. If storage is allocated for a table in a routine other than the one in which its entries are made, the name of the routine making the table entries appears in parentheses. Variables beginning with "N" are dimensions or counters, and those beginning with "L" are pointers.

DATA STORAGE

PCAM stores the data read by a READ command in arrays CDAT and C2DAT in a structure parallel to the way the data are stored in DATABASE (see Sec. II, in particular Fig. 1). For each precinct, a constant-size area of storage in CDAT contains certain data that describe the precinct as a whole, then a variable-size area contains data for each day, a constant-size area contains data for the day as a whole, and two variable-size areas contain data for each tour and each block (the size of each area depends on the number of tours read for a day). C2DAT contains data that are used to smooth the performance measures over time and data that describe the dispatch policy. The data are organized using a structure parallel to CDAT.

Each element of precinct, day, tour, and block data is referenced by a pointer that is the subscript within CDAT or C2DAT of the data for the precinct, day, tour, or block, plus an *offset* that corresponds to

Table 2

VARIABLES IN COMMON/PNTRS/

Name	Contents	Where Set (entered)
NPCTDT	Number of precincts in the base.	INIT
NPCTRD	Number of precincts read by the last READ command.	READ
LPCTDT	Pointer to subscript in CDAT of data read by last READ command.	READ
LNMLST(1)	Pointer to list of day names in current command qualifier (stored one character to a word, eight characters to a name).	GTDSPC
LNMLST(2)	Pointer to list of tour names in current command qualifier.	GTDSPC
LNMLST(3)	Pointer to list of division names in current command qualifier.	GTDSPC
LNMLST(4)	Pointer to list of precinct names in current command qualifier.	GTDSPC
NNAMES(1)	Number of day names in current command qualifier.	GTDSPC
NNAMES(2)	Number of tour names in current command qualifier.	GTDSPC
NNAMES(3)	Number of division names in current command qualifier.	GTDSPC
NNAMES(4)	Number of precinct names in current command qualifier.	GTDSPC
NDAYDT	Number of days of data in the database for each precinct.	INIT
LDAYNM	Pointer to table of names of all days in the database (8*NDAYDT words). These are in the same order as the day data for each precinct in the database.	INIT

Table 2--continued

Name	Contents	Where Set (entered)
LDYRFL	Pointer to table of day "read" flags (NDAYDT words). Each entry corresponds to one day in the database. An entry value of zero indicates that no data are to be read for that day. A nonzero value indicates that data are to be read. If the value is nonzero, then it is the ordinal position of that day among days read. It there are three days' data for each precinct in the database, and the user selects the first and third in a READ command, then the entries in this table will be 1, 0, 2.	INIT(READ)
NDAYRD	Number of days of data selected in the last READ command qualifier.	READ
LDYWFL	Pointer to table of day "work" flags (NDAYRD words). Each entry corresponds to one day for which data have been read. An entry value of zero indicates that the current command will not operate on data for that day. A nonzero value indicates that the day is to be included in the command scope. If the entry is nonzero, then it is the ordinal position of the selected day among all days in the database. Continuing the above example, if the user selects the second of the days read in a command, then the entries in this table would be 0, 3.	READ(SETWFL)
NTRDT	Number of tours in the database for each day.	INIT

Table 2--continued

Name	Contents	Where Set (entered)
LTRTB(1)	Pointer to table of blocks (NTRDT words). Each entry corresponds to a tour, the order being the same as in the database. Entry values are the ordinal position among blocks of the first block in a tour.	INIT
LTRTB(2)	The same as LTRTB(1), except gives the position of the second block for each tour. A zero-valued entry indicates that there is no second block for the tour.	INIT
LTRST	Pointer to starting hours of tours (NTRDT words). Each entry corresponds to a tour. The value of each entry is the starting hour (1-24) of that tour.	INIT
LTREND	The same as LTRST, but ending hours.	INIT
LTRRFL	Pointer to table of tour "read" flags. This is the same as LDYRFL, but for tours.	INIT(READ)
LTRNM	Pointer to table of tour names (8*NTRDT words). These are in the same order as the tour data for each day in the database.	INIT
NTRRD	Number of tours selected in the last READ command qualifier.	READ
LTRWFL	Pointer to table of tour "work" flags (NTRRD words). This is the same as LDYWFL, but for tours.	READ(SETWFL)
NBLDT	Number of blocks for each day in the database.	INIT

Table 2--continued

Name	Contents	Where Set (entered)
LBLKTB(1)	Pointer to table of starting hours for each block (NBLDT words).	INIT
LBLKTB(2)	Pointer to table of ending hours for each block (NBLDT words).	INIT
LBLRFL	Pointer to table of block "read" flags (NBLDT words). This is the same as LTRRFL, but for blocks.	INIT(READ)
NBLRD	Number of blocks read by a READ command for each day (function of the number of tours selected).	READ
LBLWFL	Not used.	
NDIVDT	Number of divisions into which precincts are aggregated.	INIT
NDIVRD	Number of divisions selected by a READ command.	READ
LDIVNM	Pointer to list of names of divisions selected by a READ command (8*NDIVDT words). Includes those selected by a request for all precincts.	INIT(READ)
LDIVFL	Pointer to list of flags that select divisions for current command (not READ) (NDIVRD words). Each entry corresponds to a division name in LDIVNM. A nonzero entry value indicates that the division was selected; a zero entry indicates that it was not.	READ(SETWFL)
IOVRLY	A flag that indicates whether there are overlay tours in the database. A value of 1 indicates that the last tour of each day in the database is an overlay tour; a value of 0 indicates that there are no overlay tours.	INIT

Table 2--continued

Name	Contents	Where Set (entered)
IOVTR(1)	If IOVRLY=1, the position of the first overlaid tour among the tours specified in a READ command. A value of m indicates that the mth tour of the tours read for each day is the tour during which the overlay tour starts.	READ
IOVTR(2)	If IOVRLY=1, the position of the second overlaid tour among the tours specified in a READ command. A value of m indicates that the mth tour of the tours read for each day is the tour during which the overlay tour ends (of course, IOVTR(2)=IOVTR(1)+1).	READ

the types of data being referenced. For example, the word containing the area of a precinct is referenced in the program by CDAT(LPCT+ARPOFF), where LPCT is the previously determined pointer to the data for the precinct and ARPOFF is the relative position within precinct data (for all precincts) of the word containing the precinct's area. Tables 3, 4, 5, and 6 give the layout of the constant data for precincts, days, tours, and blocks. LPCT, LDAY, LTOUR, and LBLK are pointers to particular precincts, days, tours, and blocks, respectively.

The offsets for all data items described above are contained in COMMON/OFFSET/. Other variables in COMMON/OFFSET/ give the storage requirements for precincts, days, tours, and blocks. These are described in Table 7.

Table 3

DESCRIPTION OF PRECINCT DATA

Data Item	Mode	Reference	Offset Value
Name (8 words)	Character	(LPCT+NMPOFF)	0
Division number (relative position in LDIVNM of division name: 0 for none)	Integer	(LPCT+DVPOFF)	8
Area (square miles)	Real	(LPCT+ARPOFF)	9
Total street length (miles)	Real	(LPCT+SMPOFF)	10
B1 (unavailability parameter)	Real	(LPCT+B1POFF)	11
B2 (unavailability parameter)	Real	(OPCT+B2POFF)	12
Data for days		(LPCT+DYPOFF)	13

Table 4

DESCRIPTION OF DAY DATA

Data Item in CDAT	Data Item in C2DAT	Mode	Reference	Offset Value
Call rate parameter		Real	(LDAY+CPDOFF)	0
Service time parameter		Real	(LDAY+SPDOFF)	1
Overlay indicator for day		Integer	(LDAY+OVDOFF)	2
Hourly call rates (24 words)	Probability of delay in previous hour	Real	(LDAY+CRDOFF)	3
Hourly service times (24 words)	Effective cars in previous hour	Real	(LDAY+STDOFF)	27
Data for tours		[a]	(LDAY+TRDOFF)	Depends on number of tours
Data for blocks		[b]	(LDAY+BLDOFF)	51

[a] See Table 5.
[b] See Table 6.

Table 5

DESCRIPTION OF TOUR DATA

Data Item in CDAT	Mode	Data Item in C2DAT	Mode	Reference	Offset Value
Difference in objective function value per car-hour if one car is added to tour	Real	Fraction of priority 1 calls dispatched 1 car	Real	(LTOUR+QDTOFF)	0
Difference in objective function value per car-hour if a car is removed from an overlay tour and one car is added to each of the tours that it overlays	Real	Fraction of priority 1 calls dispatched 2 cars	Real	(LTOUR+QXTOFF)	1
Number of calls during tour	Real	Fraction of priority 1 calls dispatched 3 cars	Real	(LTOUR+CRTOFF)	2
Objective function value with current allocation	Real	Fraction of priority 2 calls dispatched 1 car	Real	(LTOUR+QOTOFF)	3
Objective function value with one more car	Real	Fraction of priority 2 calls dispatched 2 cars	Real	(LTOUR+QNTOFF)	4
Number of most limiting constraint	Integer	Fraction of priority 2 calls dispatched 3 cars	Real	(LTOUR+CTTOFF)	5
Tour type (1=ignore, 2=standard, 3=first in overlay, 4=second in overlay, 5=overlay tour)	Integer	Fraction of priority 3 calls dispatched 1 car	Real	(LTOUR+TYTOFF)	6
Actual cars assigned to start tour	Real	Fraction of priority 3 calls dispatched 2 cars	Real	(LTOUR+ACTOFF)	7
Response speed (mph)	Real	Fraction of priority 3 calls dispatched 3 cars	Real	(LTOUR+RVTOFF)	8
Patrol speed (mph)	Real			(LTOUR+PVTOFF	9
Fraction of priority 1 calls	Real			(LTOUR+HFTOFF)	10
Fraction of priority 2 calls	Real			(LTOUR+MFTOFF)	11
Fraction of priority 3 calls	Real			(LTOUR+LFTOFF)	12

Table 6

DESCRIPTION OF BLOCK DATA

Data Item	Mode	Reference	Offset
Effective cars (including overlay effects)	Real	(LBLK+EFBOFF)	0
Actual cars on duty (including overlays)	Real	(LBLK+ACBOFF)	1
Average workload during hours of block (hours of servicing calls per hour)	Real	(LBLK+AWBOFF)	
Total calls during block	Real	(LBLK+CRBOFF)	3
Maximum workload over all hours of block	Real	(LBLK+RMBOFF)	4
Number of most limiting constraint	Real	(LBLK+CTBOFF)	6
Objective function value with current allocation	Real	(LBLK+QOBOFF)	7
Objective function value with one additional car	Real	(LBLK+QNBOFF)	8

Table 7

OTHER CONTENTS OF COMMON/OFFSET/

Variable	Contents	Value
NWDBL	Number of words required for a block	9
NWDTR	Number of words required for a tour	13
NWDDY	Number of words required for a day	51+NWDTR*NTRRD +NWDBL*NBLRD
NPRIO	Number of priority classes	3
NWDPCT	Number of words required for a precinct	13+NWDDY*NDAYRD

V. LISTING AND DESCRIPTION OF THE PCAM FORTRAN PROGRAM

The discussions in this section assume the reader's familiarity with the contents of Sec. IV (Internal Data Structures) and the User's Manual. Refer to App. C for a cross-reference listing of program segments and common blocks. The MAIN program is discussed first. Then subprograms are discussed in alphabetical order.

MAIN PROGRAM

The MAIN program primarily controls the execution of the subroutines that carry out the various PCAM commands. It operates in a continuous loop, determining which subroutine to call by examining successive command identifiers, until an END command is encountered.

Execution begins with a call to subroutine INIT to initialize permanent tables etc. Then, if operating in interactive mode, a message prompting for the user's next command is written. Subroutine SCAN is called to obtain the command identifier. If the identifier is valid, the appropriate subroutine is called to complete command interpretation and execution. When command execution is completed, the MAIN program proceeds to the next command.

The listing provided here is for a batch program. To convert to an interactive program, three clearly indicated changes must be made:

1. Remove the comment "C" on cards 2533 and 2534. This will cause the program to prompt for the next command.
2. Remove lines 2551 and 2552. These cause the printer to eject to a new page after displaying tables of output.
3. Remove line 2561. This causes page ejection after listing data.

```
      COMMON/KEYWDS/NKYWD,NTYPES,TYPOFF(4),KEYWD(8,30),WDTYPE(30)        2507
      INTEGER TYPOFF,WDTYPE                                              2508
      DIMENSION PCLSNM(8),DCLSNM(8),TOURNM(8)                            2509
      EQUIVALENCE (PCLSNM,KEYWD(1,4)),(DCLSNM,KEYWD(1,3)),               2510
     1(TOURNM,KEYWD(1,2))                                                2511
C                                                                        2512
```

```
      COMMON/SYSTEM/SYSIN,SYSOUT,IFILE,LIT                                2513
      INTEGER SYSIN,SYSOUT                                                2514
C                                                                         2515
      COMMON/SCODES/SEND,CMD,NUMLST,NAMLST,FSPEC,DSPEC,DUM,ERR            2516
      INTEGER SEND,CMD,FSPEC,DSPEC,DUM,ERR                                2517
C                                                                         2518
C                                                                         2519
      COMMON/STORE/TOP,BOT,RDBOT,MAXBOT,NWORDS,CDAT(6000),C2DAT(6000)     2520
      INTEGER TOP,BOT,RDBOT                                               2521
      DIMENSION ICDAT(6000),IC2DAT(6000)                                  2522
      EQUIVALENCE(ICDAT,CDAT),(IC2DAT,C2DAT)                              2523
C                                                                         2524
C     .                                                                   2525
      INTEGER TYPE,VAL                                                    2526
      DIMENSION VAL(2)                                                    2527
      BOT=1                                                               2528
      TOP=NWORDS+1                                                        2529
      CALL INIT                                                           2530
10    LGETT=TOP                                                           2531
C     *****  NEXT 2 LINES NEEDED FOR INTERACTIVE MODE  *****              2532
C       WRITE(SYSOUT,1)                                                   2533
C1     FORMAT(/' COMMAND? ')                                              2534
      TYPE=SEND                                                           2535
      CALL SCAN(TYPE,VAL)                                                 2536
      IF(TYPE .EQ. CMD) GO TO 20                                          2537
      WRITE(SYSOUT,2)                                                     2538
2     FORMAT(/' ***INVALID COMMAND - REENTER.')                           2539
      TOP=LGETT                                                           2540
      GO TO 10                                                            2541
20    ICMD=VAL(1)-TYPOFF(CMD)                                             2542
      GO TO (100,200,300,400,500,550,600,700,800,900),ICMD                2543
C                                                                         2544
100   CALL ADDALC(2)                                                      2545
      GO TO 10                                                            2546
200   CALL ADDALC(0)                                                      2547
      GO TO 10                                                            2548
300   CALL DISP                                                           2549
C     ***** REMOVE NEXT TWO LINES FOR INTERACTIVE MODE *****              2550
      WRITE(SYSOUT,3)                                                     2551
3     FORMAT(1H1)                                                         2552
      GO TO 10                                                            2553
400   WRITE(SYSOUT,4) MAXBOT                                              2554
4     FORMAT(/' MAXIMUM SIZE OF CURRENT-DATA WAS ',I5,' WORDS')           2555
      STOP                                                                2556
500   CALL HEAD                                                           2557
      GO TO 10                                                            2558
550    CALL LIST                                                          2559
C     ***** REMOVE NEXT LINE FOR INTERACTIVE MODE *****                   2560
      WRITE(SYSOUT,3)                                                     2561
       GO TO 10                                                           2562
600   CALL MEET                                                           2563
      GO TO 10                                                            2564
```



```
700    CALL READ                                                          2565
      GO TO 10                                                            2566
800    CALL SET                                                           2567
       GO TO 10                                                           2568
900   CALL WRITE                                                          2569
      GO TO 10                                                            2570
      END                                                                 2571
```

SUBROUTINE ADDALC

Subroutine ADDALC (*add* and *al*locate) carries out the ADD and ALLOCATE commands. Its parameter ISW determines which command is to be executed. If ISW is less than 2, then the ALOC command is executed; otherwise, the ADD command is executed.

Successive calls to subroutine SCAN get the user's specification of the number of car-hours; subroutine GTDSPC scans the command qualifier; and additional calls to SCAN get the user's objective function specification. The user's specification of the number of car-hours is saved as follows: NHOURS holds the numeric part of any specification; ISTAR is 1, 0, or -1, depending upon whether the user's expression is of the form *-n, n, or n-* (* alone is equivalent to *-0).

If an asterisk appears in the expression giving the number of car-hours, the number of car-hours currently allocated to all selected shifts is determined and the expression is evaluated to give a number of car-hours to be allocated or added.

The program then indexes through all selected precincts and days. If an ALOC command is being executed, each block of each selected tour of each day is assigned just enough cars to handle its cfs workload, and subroutines STRCAR, SBLACT, and SBLEF are called to get a feasible allocation of cars to tours and to translate the tour allocation back to a block allocation; this step is skipped for ADD commands. The objective function is evaluated for each selected block of a day by means of a call to SBLOBJ and for each selected tour of a day via a call to STROBJ. The constraint indicators for each block of selected tours are set to zero. Subroutine ADJUST is called for ALOC commands to insure that the initial allocation results in the minimum objective function value for the number of car-hours assigned in each day.

After the objective function has been evaluated for all shifts, the number of car-hours that remain to be allocated is computed, and subroutine ADDCAR is called to allocate the number of car-hours.


```
      SUBROUTINE ADDALC(ISW)
C
C  PERFORMS ADD OR ALLOCATE FUNCTION, DEPENDING ON
C THE VALUE OF 'ISW'.
C
C
C
      COMMON/STORE/TOP,BOT,RDBOT,MAXBOT,NWORDS,CDAT(6000),C2DAT(6000)
      INTEGER TOP,BOT,RDBOT
      DIMENSION ICDAT(6000),IC2DAT(6000)
      EQUIVALENCE(ICDAT,CDAT),(IC2DAT,C2DAT)
C
C
C
      COMMON/SYSTEM/SYSIN,SYSOUT,IFILE,LIT
      INTEGER SYSIN,SYSOUT
C
      COMMON/PNTRS/IOVRLY,IOVTR(2),
     1NPCTDT,NPCTRD,LPCTDT,LNMLST(4),NNAMES(4),NDAYDT,LDAYNM,
     2LDYRFL,NDAYRD,LDYWFL,NTRDT,LTRTB(2),LTRST,LTREND,LTRRFL,LTRNM,
     3NTRRD,LTRWFL,NBLDT,LBLKTB(2),LBLRFL,NBLRD,LBLWFL,NDIVDT,NDIVRD,
     4LDIVNM,LDIVFL
C
      COMMON/OFFSET/NMPOFF,DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,
     1NWDPCT,CPDOFF,SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,NWDDY,
     2QDTOFF,QXTOFF,CRTOFF,QOTOFF,QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,
     3PVTOFF,HFTOFF,MFTOFF,LFTOFF,NPRIO,NWDTR,BLDOFF,QOBOFF,QNBOFF,
     4EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,NWDBL
C
      INTEGER DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,CPDOFF,
     1SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,QDTOFF,QXTOFF,CRTOFF,QOTOFF,
     2QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,PVTOFF,HFTOFF,BLDOFF,
     3EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,QOBOFF,QNBOFF
C
      COMMON/KEYWDS/NKYWD,NTYPES,TYPOFF(4),KEYWD(8,30),WDTYPE(30)
      INTEGER TYPOFF,WDTYPE
      DIMENSION PCLSNM(8),DCLSNM(8),TOURNM(8)
      EQUIVALENCE (PCLSNM,KEYWD(1,4)),(DCLSNM,KEYWD(1,3)),
     1(TOURNM,KEYWD(1,2))
C
      COMMON/SCODES/SEND,CMD,NUMLST,NAMLST,FSPEC,DSPEC,DUM,ERR
      INTEGER SEND,CMD,FSPEC,DSPEC,DUM,ERR
C
      DIMENSION VAL(2),ORDER(3),EN(4)
      INTEGER TYPE,VAL
C
      INTEGER CHARST,CHARMN
      DATA CHARST/1H*/,CHARMN/1H-/
C
      ISMFLG = IC2DAT(1)
      LGETT=TOP
      TYPE=CMD
```

```
      NHOURS=0                                                          53
      ISTAR=0                                                           54
C                                                                       55
C          GET EXPRESSION FOR CAR HOURS TO ALLOCATE                     56
C                                                                       57
      CALL SCAN(TYPE,VAL)                                               58
      IF(TYPE .EQ. NUMLST) GO TO 20                                     59
      IF(TYPE .EQ. NAMLST) GO TO 15                                     60
10    WRITE(SYSOUT,1)                                                   61
1     FORMAT(/' *** INVALID NUMBER OF CAR HOURS TO ALLOCATE - ',        62
     1 'REENTER')                                                       63
      TOP=LGETT                                                         64
      RETURN                                                            65
15    IF(ICDAT(VAL(2)) .NE. CHARST) GO TO 10                            66
      ISTAR=1                                                           67
      CALL SCAN(TYPE,VAL)                                               68
      IF(TYPE .NE. NUMLST) GO TO 25                                     69
      NHOURS=ICDAT(VAL(2))                                              70
      IF(NHOURS .GT. 0) GO TO 10                                        71
      CALL SCAN(TYPE,VAL)                                               72
      GO TO 25                                                          73
20    NHOURS=ICDAT(VAL(2))                                              74
      CALL SCAN(TYPE,VAL)                                               75
      IF(TYPE .NE. NAMLST) GO TO 25                                     76
      IF(ICDAT(VAL(2)) .NE. CHARMN .OR. ICDAT(VAL(2)+1) .NE. CHARST)    77
     1 GO TO 10                                                         78
      ISTAR=-1                                                          79
      CALL SCAN(TYPE,VAL)                                               80
C                                                                       81
C          SCAN QUALIFIER                                               82
C                                                                       83
25    CALL GTDSPC(TYPE,VAL,ORDER)                                       84
      IF(TYPE .NE. ERR) GO TO 30                                        85
      TOP=LGETT                                                         86
      RETURN                                                            87
C                                                                       88
C          SCAN AND VALIDATE OBJECTIVE FUNCTION SPECIFICATION           89
C                                                                       90
30    IF(TYPE .EQ. FSPEC) GO TO 60                                      91
50    WRITE(SYSOUT,2)                                                   92
2     FORMAT(/' ***INVALID OBJECTIVE FUNCTION - REENTER')               93
      TOP=LGETT                                                         94
      RETURN                                                            95
60    KEYOFF=VAL(1)                                                     96
      I=KEYOFF-TYPOFF(FSPEC)                                            97
      IF(I .NE. 4) GO TO 50                                             98
      CALL SCAN(TYPE,VAL)                                               99
      IF(TYPE .NE. NUMLST) GO TO 50                                    100
      NPARM=VAL(1)                                                     101
      LPARM=VAL(2)                                                     102
      IFNCTN=ICDAT(LPARM)                                              103
      IF(IFNCTN .LT. 1 .OR. IFNCTN .GT. 3) GO TO 50                    104
```



```
C                                                                       105
      IF(IFNCTN .EQ. 1) GO TO 130                                       106
C                                                                       107
      IF(NPARM .GT. 1) GO TO 70                                         108
      CALL GETTOP(4,LPARM)                                              109
C                                                                       110
      ICDAT(LPARM)=IFNCTN                                               111
      ICDAT(LPARM+2)=0                                                  112
      GO TO 130                                                         113
70    IPRIO=ICDAT(LPARM+2)                                              114
      IF(IPRIO .GE. 0 .AND. IPRIO .LE. NPRIO) GO TO 130                 115
      WRITE(SYSOUT,3)                                                   116
3     FORMAT(/' ***INVALID OBJECTIVE FUNCTION PARAMETER(S) - REENTER')  117
      TOP=LGETT                                                         118
      RETURN                                                            119
C                                                                       120
C          SET WORK FLAGS                                               121
C                                                                       122
130   CALL SETWFL(IERR)                                                 123
      IF(IERR .EQ. 0) GO TO 132                                         124
      TOP=LGETT                                                         125
      RETURN                                                            126
C                                                                       127
C          CHECK OVERLAY SPECIFICATION                                  128
C                                                                       129
132   CALL CKOVR(IERR)                                                  130
      IF(IERR .EQ. 0) GO TO 135                                         131
      TOP=LGETT                                                         132
      RETURN                                                            133
135   NCRHRS=0                                                          134
C                                                                       135
C ** TO ALLOW THE USER TO SPECIFY CARS TO ALLOCATE INSTEAD OF CAR HOURS 136
C ** A STATEMENT SHOULD BE ADDED HERE TO MULTIPLY 'NHOURS' BY THE LENGTH 137
C ** OF A TOUR; E.G. INSERT THE STATEMENT NHOURS=NHOURS*8 IF ALL TOURS  138
C ** ARE EIGHT HOURS IN LENGTH AND CARS (INSTEAD OF CAR HOURS)          139
C                                                                       140
      IF(ISTAR .EQ. 0) GO TO 180                                        141
C                                                                       142
C          FIND NUM OF CAR HOURS ALREADY ASSIGNED TO SELECTED SHIFTS    143
C                                                                       144
      CRHRS=NCRHRS                                                      145
      LPCT=0                                                            146
140   LPCT=NXPCT(LPCT)                                                  147
      IF(LPCT .EQ. 0) GO TO 170                                         148
      LDAY=0                                                            149
150   LDAY=NXDAY(LPCT,LDAY)                                             150
      IF(LDAY .EQ. 0) GO TO 140                                         151
      LTOUR=0                                                           152
160   LTOUR=NXTOUR(LDAY,LTOUR,ITYPE)                                    153
      IF(LTOUR .EQ. 0) GO TO 150                                        154
      TOURLN=ICDAT(LTREND+ITYPE-1)-ICDAT(LTRST+ITYPE-1)+1               155
                                                                        156
```

```
      CRHRS=CRHRS+CDAT(LTOUR+ACTOFF)*TOURLN                              157
      GO TO 160                                                          158
C                                                                        159
C           COMPUTE CAR HOURS AVAILABLE TO ALLOCATE                      160
C                                                                        161
170   NCRHRS=INT(CRHRS+.5)                                               162
      NHOURS=NHOURS+ISTAR*NCRHRS                                         163
C                                                                        164
C           DETERMINE INITIAL ASSIGNMENT (ALOC ONLY) AND EVALUATE        165
C           CORRESPONDING OBJECTIVE FUNCTION VALUES FOR ALL SHIFTS       166
C                                                                        167
180   NTOT=0                                                             168
      LPCT=0                                                             169
200   LPCT=NXPCT(LPCT)                                                   170
      IF(LPCT .EQ. 0) GO TO 300                                          171
      B1=CDAT(LPCT+B1POFF)                                               172
      B2=CDAT(LPCT+B2POFF)                                               173
      LDAY=0                                                             174
210   LDAY=NXDAY(LPCT,LDAY)                                              175
      IF(LDAY .EQ. 0) GO TO 200                                          176
      IF(ISW .GT. 1) GO TO 245                                           177
C                                                                        178
C           FIND MINIMUM ASSIGNMENT FOR EACH BLOCK                       179
C                                                                        180
      IBL=0                                                              181
      LTOUR=0                                                            182
220   LTOUR=NXTOUR(LDAY,LTOUR,ITYPE)                                     183
      IF(LTOUR .EQ. 0) GO TO 240                                         184
      IND=ICDAT(LTOUR+TYTOFF)                                            185
      IF(IND .EQ. 5) GO TO 240                                           186
      DO 230 IBLK=1,2                                                    187
      IBDT=ICDAT(LTRTB(IBLK)+ITYPE-1)                                    188
      IF(IBDT .LT. 1) GO TO 230                                          189
      IBRD=ICDAT(LBLRFL+IBDT-1)                                          190
      LBLK=LDAY+BLDOFF+(IBRD-1)*NWDBL                                    191
      AWL=CDAT(LBLK+AWBOFF)                                              192
C                                                                        193
      EF= INT(CDAT(LBLK+RMBOFF)+1.0001)                                  194
C                                                                        195
      IF(ISMFLG .EQ. 1)                                                  196
     1 EF=INT(CDAT(LBLK+AWBOFF)+1.0001)                                  197
      ACT=CEIL((EF+B1*AWL)/(1.-B2))                                      198
      EF=ACT*(1.-((B1*AWL/ACT)+B2))                                      199
      CDAT(LBLK+ACBOFF)=ACT                                              200
      CDAT(LBLK+EFBOFF)=EF                                               201
      IF(IND .NE. 3 .AND. IND .NE. 4) GO TO 230                          202
      IBL=IBL+1                                                          203
      EN(IBL)=ACT                                                        204
230   CONTINUE                                                           205
      GO TO 220                                                          206
C                                                                        207
C           FIND MINIMUM FEASIBLE TOUR ASSIGNMENT                        208
```



```
C                                                                        209
240   CALL STRCAR(LDAY,CARHRS)                                           210
      NTOT=NTOT+CARHRS                                                   211
      CALL SBLACT(LPCT,LDAY)                                             212
      CALL SBLEF(LPCT,LDAY)                                              213
245   LTOUR=0                                                            214
250   LTOUR=NXTOUR(LDAY,LTOUR,ITYPE)                                     215
      IF(LTOUR .NE. 0) GO TO 255                                         216
      IF(IOVRLY .EQ. 0 .OR. ICDAT(LDAY+OVDOFF) .EQ. 0 .OR. ISW .GT. 1)   217
     C GO TO 210                                                         218
      IF(ICDAT(LTRWFL+NTRRD-1) .EQ. 0) GO TO 210                         219
      LSTOUR=LDAY+TRDOFF+(NTRRD-1)*NWDTR                                 220
      IF(ICDAT(LSTOUR+TYTOFF) .NE. 5) GO TO 210                          221
C                                                                        222
C           ADJUST INITIAL TOUR ASSIGNMENT TO MINIMIZE                   223
C           OBJECTIVE FUNCTION                                           224
C                                                                        225
      X1=AMAX1(EN(2)-EN(1),0.)                                           226
      X2=AMAX1(EN(3)-EN(4),0.)                                           227
      DELTA=X1-X2                                                        228
      CALL ADJUST(LPARM,LPCT,LDAY,DELTA)                                 229
      GO TO 210                                                          230
255   ICDAT(LTOUR+CTTOFF)=0                                              231
C                                                                        232
C           COMPUTE INITIAL OBJECTIVE FUNCTION VALUES FOR BLOCKS         233
C                                                                        234
      DO 260 IBLK=1,2                                                    235
      IBDT=ICDAT(LTRTB(IBLK)+ITYPE-1)                                    236
      IF(IBDT .LT. 1) GO TO 260                                          237
      IBRD=ICDAT(LBLRFL+IBDT-1)                                          238
      LBLK=LDAY+BLDOFF+(IBRD-1)*NWDBL                                    239
      ICDAT(LBLK+CTBOFF)=0                                               240
      IF(ICDAT(LTOUR+TYTOFF) .EQ. 5) GO TO 260                           241
      CDAT(LBLK+ACBOFF)=CDAT(LBLK+ACBOFF)-2.                             242
      CALL SBLOBJ(LPARM,LPCT,LDAY,LTOUR,LBLK,IBDT)                       243
      CALL SBLOBJ(LPARM,LPCT,LDAY,LTOUR,LBLK,IBDT)                       244
260   CONTINUE                                                           245
      CALL STROBJ(LDAY,LTOUR,ITYPE)                                      246
      GO TO 250                                                          247
C                                                                        248
C           ALLOCATE REMAINING CAR HOURS                                 249
C                                                                        250
300   IF(ISW .LT. 2) GO TO 305                                           251
      NLEFT=NHOURS                                                       252
      IF(NLEFT .GT. 0) GO TO 310                                         253
      RETURN                                                             254
305   NLEFT=NHOURS-NTOT                                                  255
      IF(NLEFT .GE. 0) GO TO 310                                         256
      WRITE(SYSOUT,4) NTOT                                               257
4     FORMAT(/' *** ',I5,' CAR HOURS ALLOCATED.')                        258
      TOP=LGETT                                                          259
      RETURN                                                             260
```

```
C                                                                       261
310     CALL ADDCAR(NLEFT,LPARM)                                        262
        TOP=LGETT                                                       263
        RETURN                                                          264
        END                                                             265
```


SUBROUTINE ADDCAR

Subroutine ADDCAR (*add cars*) adds cars to a set of shifts so that the average value of a specified objective function is minimized. Parameter LPARM is a pointer to a number list that specifies the function to be evaluated. NCARHR is the number of car-hours available for allocation.

The allocation algorithm used is described in App. A of the User's Manual.

```
        SUBROUTINE ADDCAR(NCARHR,LPARM)                                 266
C                                                                       267
C  ADDS CARS TO A SET OF SHIFTS SO THAT THE AVERAGE VALUE               268
C  OF A SPECIFIED OBJECTIVE FUNCTION IS MINIMIZED                       269
C                                                                       270
        COMMON/STORE/TOP,BOT,RDBOT,MAXBOT,NWORDS,CDAT(6000),C2DAT(6000) 271
        INTEGER TOP,BOT,RDBOT                                           272
        DIMENSION ICDAT(6000),IC2DAT(6000)                              273
        EQUIVALENCE(ICDAT,CDAT),(IC2DAT,C2DAT)                          274
C                                                                       275
C                                                                       276
        COMMON/PNTRS/IOVRLY,IOVTR(2),                                   277
     1NPCTDT,NPCTRD,LPCTDT,LNMLST(4),NNAMES(4),NDAYDT,LDAYNM,           278
     2LDYRFL,NDAYRD,LDYWFL,NTRDT,LTRTB(2),LTRST,LTREND,LTRRFL,LTRNM,    279
     3NTRRD,LTRWFL,NBLDT,LBLKTB(2),LBLRFL,NBLRD,LBLWFL,NDIVDT,NDIVRD,   280
     4LDIVNM,LDIVFL                                                     281
C                                                                       282
        COMMON/OFFSET/NMPOFF,DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF, 283
     1NWDPCT,CPDOFF,SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,NWDDY,           284
     2QDTOFF,QXTOFF,CRTOFF,QOTOFF,QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,   285
     3PVTOFF,HFTOFF,MFTOFF,LFTOFF,NPRIO,NWDTR,BLDOFF,QOBOFF,QNBOFF,     286
     4EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,NWDBL            287
C                                                                       288
        INTEGER DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,CPDOFF,       289
     1SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,QDTOFF,QXTOFF,CRTOFF,QOTOFF,   290
     2QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,PVTOFF,HFTOFF,BLDOFF,          291
     3EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,QOBOFF,QNBOFF    292
C                                                                       293
        COMMON/SYSTEM/SYSIN,SYSOUT,IFILE,LIT                            294
        INTEGER SYSIN,SYSOUT                                            295
C                                                                       296
        NLEFT=NCARHR                                                    297
C                                                                       298
C           FIND SHIFT WITH GREATEST IMPROVEMENT PER CAR HOUR IN        299
C           OBJECTIVE FUNCTION VALUE IF ALLOCATION IS CHANGED           300
C           INCREMENTALLY.                                              301
C                                                                       302
310     LBPCT=NXPCT(0)                                                  303
                                                                        304
```

```
      LBDAY=NXDAY(LBPCT,0)
      LBTOUR=NXTOUR(LBDAY,0,IBTYPE)
      QBIG=CDAT(LBTOUR+QDTOFF)
      LPCT=0
320   LPCT=NXPCT(LPCT)
      IF(LPCT .EQ. 0) GO TO 350
      LDAY=0
330   LDAY=NXDAY(LPCT,LDAY)
      IF(LDAY .EQ. 0) GO TO 320
      LTOUR=0
340   LTOUR=NXTOUR(LDAY,LTOUR,ITYPE)
      IF(LTOUR .EQ. 0) GO TO 330
      QDIF=AMAX1(CDAT(LTOUR+QDTOFF),CDAT(LTOUR+QXTOFF))
      IF(QDIF .LE. QBIG) GO TO 340
      QBIG=QDIF
      IBTYPE=ITYPE
      LBTOUR=LTOUR
      LBDAY=LDAY
      LBPCT=LPCT
      GO TO 340
C
350   IF(LBTOUR .NE. 0) GO TO 360
      WRITE(SYSOUT,5)
5     FORMAT(/' *** NO SHIFTS SELECTED - REENTER')
      RETURN
360   IF(ICDAT(LBTOUR+TYTOFF) .EQ. 5 .AND. CDAT(LBTOUR+QXTOFF)
     C      .GT. CDAT(LBTOUR+QDTOFF)) GO TO 500
      ILEN=ICDAT(LTREND+IBTYPE-1)-ICDAT(LTRST+IBTYPE-1)+1
      IF(ILEN .GT. NLEFT) RETURN
C
C          ADD A CAR TO SELECTED SHIFT AND COMPUTE NEW OBJECTIVE
C          FUNCTION VALUE
C
      CDAT(LBTOUR+ACTOFF)=CDAT(LBTOUR+ACTOFF)+1.
      NLEFT=NLEFT-ILEN
      CDAT(LBTOUR+QOTOFF)=CDAT(LBTOUR+QNTOFF)
      CDAT(LBTOUR+QNTOFF)=0.
      DO 370 IB=1,2
      IBDT=ICDAT(LTRTB(IB)+IBTYPE-1)
      IF(IBDT .LT. 1) GO TO 370
      IBRD=ICDAT(LBLRFL+IBDT-1)
      LBLK=LBDAY+BLDOFF+(IBRD-1)*NWDBL
      LTTOUR=LBTOUR
      IF(ICDAT(LBTOUR+TYTOFF) .EQ. 5)
     1  LTTOUR=LBDAY+TRDOFF+(IOVTR(IB)-1)*NWDTR
      CALL SBLOBJ(LPARM,LBPCT,LBDAY,LTTOUR,LBLK,IBDT)
      CDAT(LBTOUR+QNTOFF)=CDAT(LBTOUR+QNTOFF)+CDAT(LBLK+QNBOFF)
370   CONTINUE
      CALL STRDF(LBDAY,LBTOUR,IBTYPE)
      ID=ICDAT(LBTOUR+TYTOFF)-1
      GO TO (310,390,390,410),ID
C
```



```
C
C          ADJUST OBJECTIVE FUNCTION DIFFERENCES FOR SHIFTS IN
C          OVERLAY SEGMENTS
C
390   ITRRD=ICDAT(LTRRFL+NTRDT-1)
      LTTOUR=LBDAY+TRDOFF+(ITRRD-1)*NWDTR
      CALL STROBJ(LBDAY,LTTOUR,NTRDT)
      GO TO 310
C
410   DO 420 I=1,2
      ITRRD=IOVTR(I)
      ITYPE=ICDAT(LTRWFL+ITRRD-1)
      LTTOUR=LBDAY+TRDOFF+(ITRRD-1)*NWDTR
420   CALL STROBJ(LBDAY,LTTOUR,ITYPE)
      GO TO 310
C
C          DECREASE OVERLAY SHIFT ASSIGNMENT AND INCREASE ASSIGNMENTS
C          TO OVERLAID SHIFTS
C
500   ITOT=0
      DO 510 I=1,2
      ITRRD=IOVTR(I)
      ITP=ICDAT(LTRWFL+ITRRD-1)
      ISTART=ICDAT(LTRST+ITP-1)
      IEND=ICDAT(LTREND+ITP-1)
510   ITOT=ITOT+IEND-ISTART+1
      ILEN=ITOT-(ICDAT(LTREND+IBTYPE-1)-ICDAT(LTRST+IBTYPE-1)+1)
      IF(ILEN .GT. NLEFT) RETURN
      NLEFT=NLEFT-ILEN
      CDAT(LBTOUR+ACTOFF)=CDAT(LBTOUR+ACTOFF)-1.
      DO 520 I=1,2
      ITRRD=IOVTR(I)
      ITP=ICDAT(LTRWFL+ITRRD-1)
      IBDT=ICDAT(LTRTB(I)+ITP-1)
      IBRD=ICDAT(LBLRFL+IBDT-1)
      LBLK=LBDAY+BLDOFF+(IBRD-1)*NWDBL
      LTTOUR=LBDAY+TRDOFF+(ITRRD-1)*NWDTR
      CDAT(LTTOUR+ACTOFF)=CDAT(LTTOUR+ACTOFF)+1.
      CALL SBLOBJ(LPARM,LBPCT,LBDAY,LTTOUR,LBLK,IBDT)
      CALL STROBJ(LBDAY,LTTOUR,ITP)
520   CONTINUE
      CALL STRDF(LBDAY,LBTOUR,IBTYPE)
      GO TO 310
      END
```

SUBROUTINE ADJUST

Subroutine ADJUST insures that the initial assignment of cars to shifts for an ALOC command results in the lowest possible objective function value. LPARM is a pointer to a parameter list that specifies the objective function. LPCT and LDAY are pointers to the data for the precinct and day. The absolute value of XDELT is the maximum number of cars that can be moved from an overlay shift to an overlaid shift to reduce the objective function value. The sign of XDELT indicates whether cars can be moved to the first overlaid shift (positive) or the second overlaid shift (negative). No cars can be shifted if XDELT is zero or no cars are assigned to the overlay shift. Up to ABS(XDELT) cars are moved from the overlay shift to the appropriate overlaid shift. The process terminates when moving another car would increase the objective function value.

```
      SUBROUTINE ADJUST(LPARM,LPCT,LDAY,XDELT)                          400
C                                                                       401
C SUBROUTINE TO EXAMIN ALTERNATIVE INITIAL ALLOCATIONS FOR THE          402
C ALOC COMMAND TO FIND THE INITIAL ALLOCATION WITH THE BEST             403
C OBJECTIVE FUNCTION VALUE                                              404
C                                                                       405
C                                                                       406
      COMMON/STORE/TOP,BOT,RDBOT,MAXBOT,NWORDS,CDAT(6000),C2DAT(6000)   407
      INTEGER TOP,BOT,RDBOT                                             408
      DIMENSION ICDAT(6000),IC2DAT(6000)                                409
      EQUIVALENCE(ICDAT,CDAT),(IC2DAT,C2DAT)                            410
C                                                                       411
C                                                                       412
      COMMON/OFFSET/NMPOFF,DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,   413
     1NWDPCT,CPDOFF,SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,NWDDY,           414
     2QDTOFF,QXTOFF,CRTOFF,QOTOFF,QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,   415
     3PVTOFF,HFTOFF,MFTOFF,LFTOFF,NPRIO,NWDTR,BLDOFF,QOBOFF,QNBOFF,     416
     4EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,NWDBL            417
C                                                                       418
      INTEGER DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,CPDOFF,         419
     1SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,QDTOFF,QXTOFF,CRTOFF,QOTOFF,   420
     2QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,PVTOFF,HFTOFF,BLDOFF,          421
     3EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,QOBOFF,QNBOFF    422
C                                                                       423
      COMMON/PNTRS/IOVRLY,IOVTR(2),                                     424
     1NPCTDT,NPCTRD,LPCTDT,LNMLST(4),NNAMES(4),NDAYDT,LDAYNM,           425
     2LDYRFL,NDAYRD,LDYWFL,NTRDT,LTRTB(2),LTRST,LTREND,LTRRFL,LTRNM,    426
     3NTRRD,LTRWFL,NBLDT,LBLKTB(2),LBLRFL,NBLRD,LBLWFL,NDIVDT,NDIVRD,   427
```



```
     4LDIVNM,LDIVFL
C                                                                       428
C                                                                       429
C          FIND BLOCKS WHOSE ASSIGNMENTS CAN BE CHANGED                 430
C                                                                       431
      LOVTR=LDAY+TRDOFF+(NTRRD-1)*NWDTR                                 432
      ENOV=CDAT(LOVTR+ACTOFF)                                           433
      IF(ENOV .LE. 0.) RETURN                                           434
      ISW=1                                                             435
      IF(XDELT .LT. 0.) ISW=2                                           436
      ITRRD=IOVTR(ISW)                                                  437
      ITYPE=ICDAT(LTRWFL+ITRRD-1)                                       438
      ILEN=ICDAT(LTREND+ITYPE-1)-ICDAT(LTRST+ITYPE-1)+1                 439
      IOVLN=ICDAT(LTREND+NTRDT-1)-ICDAT(LTRST+NTRDT-1)+1                440
      IF(ILEN .GT. IOVLN) RETURN                                        441
      DELTA=ABS(XDELT)                                                  442
      IF(DELTA .LT. .9999) RETURN                                       443
      INV=2/ISW                                                         444
      IBDT2=ICDAT(LTRTB(INV)+NTRDT-1)                                   445
      IBRD=ICDAT(LBLRFL+IBDT2-1)                                        446
      LBLK2=LDAY+BLDOFF+(IBRD-1)*NWDBL                                  447
      ISTART=ICDAT(LBLKTB(1)+IBDT2-1)                                   448
      IEND=ICDAT(LBLKTB(2)+IBDT2-1)                                     449
      IBDT1=IBDT2+2*(-1)**ISW                                           450
      IBRD=ICDAT(LBLRFL+IBDT1-1)                                        451
      LBLK1=LDAY+BLDOFF+(IBRD-1)*NWDBL                                  452
      LTOUR=LDAY+TRDOFF+(ITRRD-1)*NWDTR                                 453
      ITRRD=IOVTR(INV)                                                  454
      LTTOUR=LDAY+TRDOFF+(ITRRD-1)*NWDTR                                455
      ITTYPE=ICDAT(LTRWFL+ITRRD-1)                                      456
      B1=CDAT(LPCT+B1POFF)                                              457
      B2=CDAT(LPCT+B2POFF)                                              458
      AWL=CDAT(LBLK2+AWBOFF)                                            459
C                                                                       460
C                                                                       461
C          ADJUST BLOCK AND TOUR ASSIGNMENTS TO MINIMIZE OBJECTIVE      462
C          FUNCTION VALUE                                               463
C                                                                       464
10    QOLD=CDAT(LBLK1+QOBOFF)+CDAT(LBLK2+QOBOFF)                        465
      ACT=CDAT(LBLK2+ACBOFF)-1.                                         466
      EF=ACT*(1.-((B1*AWL/ACT)+B2))                                     467
      QTEST=OBJFUN(LPARM,ISTART,IEND,LPCT,LDAY,LTTOUR,EF)               468
      QNEW=CDAT(LBLK1+QNBOFF)+QTEST                                     469
      IF(QOLD .LT. QNEW) GO TO 100                                      470
      CALL SBLOBJ(LPARM,LPCT,LDAY,LTOUR,LBLK1,IBDT1)                    471
      CDAT(LTOUR+ACTOFF)=CDAT(LTOUR+ACTOFF)+1.                          472
      CDAT(LOVTR+ACTOFF)=CDAT(LOVTR+ACTOFF)-1.                          473
      CDAT(LBLK2+QNBOFF)=CDAT(LBLK2+QOBOFF)                             474
      CDAT(LBLK2+ACBOFF)=ACT                                            475
      CDAT(LBLK2+EFBOFF)=EF                                             476
      CDAT(LBLK2+QOBOFF)=QTEST                                          477
      DELTA=DELTA-1.                                                    478
      IF(DELTA .GT. 0.) GO TO 10                                        479
```

```
100     CALL STROBJ(LDAY,LOVTR,NTRDT)                                    480
        CALL STROBJ(LDAY,LTOUR,ITYPE)                                    481
        CALL STROBJ(LDAY,LTTOUR,ITTYPE)                                  482
        RETURN                                                           483
        END                                                              484
```


FUNCTION AVTT

Function AVTT returns the *av*erage *t*ravel *t*ime to incidents over a specified span of hours of a particular day in a precinct. Parameters ISTART and IEND give the first and last hour for which travel time is computed. LPCT and LDAY are pointers to the data for the precinct and day. RV is the response speed of patrol units and EF is the number of effective cars on duty.

```
C       FUNCTION AVTT(ISTART,IEND,LPCT,LDAY,RV,EF)                       485
C  CALCULATES AVERAGE TRAVEL TIME                                        486
C                                                                        487
C                                                                        488
        COMMON/STORE/TOP,BOT,RDBOT,MAXBOT,NWORDS,CDAT(6000),C2DAT(6000)  489
        INTEGER TOP,BOT,RDBOT                                            490
        DIMENSION ICDAT(6000),IC2DAT(6000)                               491
        EQUIVALENCE(ICDAT,CDAT),(IC2DAT,C2DAT)                           492
C                                                                        493
C                                                                        494
      COMMON/OFFSET/NMPOFF,DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,    495
     1NWDPCT,CPDOFF,SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,NWDDY,            496
     2QDTOFF,QXTOFF,CRTOFF,QOTOFF,QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,    497
     3PVTOFF,HFTOFF,MFTOFF,LFTOFF,NPRIO,NWDTR,BLDOFF,QOBOFF,QNBOFF,      498
     4EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,NWDBL             499
C                                                                        500
      INTEGER DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,CPDOFF,          501
     1SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,QDTOFF,QXTOFF,CRTOFF,QOTOFF,    502
     2QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,PVTOFF,HFTOFF,BLDOFF,           503
     3EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,QOBOFF,QNBOFF     504
C                                                                        505
      TD=0.                                                              506
      CRT=0.                                                             507
      LCR=LDAY+CRDOFF-1                                                  508
      LST=LDAY+STDOFF-1                                                  509
      A=CDAT(LPCT+ARPOFF)                                                510
      STRDNS=CDAT(LPCT+SMPOFF)/A                                         511
      G=(STRDNS-1.)/(STRDNS-2.)                                          512
      SQRTA=SQRT(A)                                                      513
      DO 30 I=ISTART,IEND                                                514
      CR=CDAT(LCR+I)                                                     515
      ST=CDAT(LST+I)                                                     516
      CRT=CRT+CR                                                         517
      AVAVL=EF-CR*ST                                                     518
C                                                                        519
C                                                                        520
C       USE TRAVEL DISTANCE FUNCTION APPROPRIATE FOR AVG                 521
C       CARS AVAILABLE IN AN HOUR                                        522
                                                                         523
```

```
      IF(AVAVL .GE. 1.) GO TO 10                                          524
      TD=TD+.678*SQRTA*CR                                                 525
      GO TO 30                                                            526
   10 IF(AVAVL .GE.2.) GO TO 20                                           527
      TD=TD+SQRTA*(.08 +.598/SQRT(AVAVL))*CR                              528
      GO TO 30                                                            529
   20 TD=TD+.711*CR*SQRTA/SQRT(AVAVL)                                     530
   30 CONTINUE                                                            531
C                                                                         532
C        COMPUTE AVERAGE TRAVEL TIME FROM TRAVEL DISTANCE WITH            533
C        STREET DENSITY CORRECTION                                        534
C                                                                         535
      AVTT=60.*G*TD/(CRT*RV)                                              536
      RETURN                                                              537
      END                                                                 538
```


FUNCTION CEIL

CEIL(X) is the least integer that is greater than or equal to X.

```
      FUNCTION CEIL(X)
C                                                                         539
C  LEAST INTEGER GREATER THAN OR EQUAL TO X                               540
C                                                                         541
      ICEIL=X                                                             542
      CEIL=ICEIL                                                          543
      IF(X.GT.CEIL)CEIL=CEIL+1.                                           544
      RETURN                                                              545
      END                                                                 546
                                                                          547
```

```
      IF(AVAVL .GE. 1.) GO TO 10                                          524
      TD=TD+.678*SQRTA*CR                                                 525
      GO TO 30                                                            526
10    IF(AVAVL .GE.2.) GO TO 20                                           527
      TD=TD+SQRTA*(.08 +.598/SQRT(AVAVL))*CR                              528
      GO TO 30                                                            529
20    TD=TD+.711*CR*SQRTA/SQRT(AVAVL)                                     530
30    CONTINUE                                                            531
C                                                                         532
C           COMPUTE AVERAGE TRAVEL TIME FROM TRAVEL DISTANCE WITH         533
C           STREET DENSITY CORRECTION                                     534
C                                                                         535
      AVTT=60.*G*TD/(CRT*RV)                                              536
      RETURN                                                              537
      END                                                                 538
```


FUNCTION CEIL

CEIL(X) is the least integer that is greater than or equal to X.

```
      FUNCTION CEIL(X)                                                    539
C                                                                         540
C  LEAST INTEGER GREATER THAN OR EQUAL TO X                               541
C                                                                         542
      ICEIL=X                                                             543
      CEIL=ICEIL                                                          544
      IF(X.GT.CEIL)CEIL=CEIL+1.                                           545
      RETURN                                                              546
      END                                                                 547
```

SUBROUTINE CKOVR

Subroutine CKOVR (check overlay) is used to insure that, if the user has selected an overlay tour in a command qualifier, then he has also selected the overlaid tours. Its parameter IERR is set to zero (0) on return if a valid specification has been made, otherwise it is set to one (1). The determination of validity is based on the "work" flags of the tours involved. See Sec. IV on table pointers for a description of the flags and tables involved.

```
      SUBROUTINE CKOVR(IERR)                                              548
C                                                                         549
C CHECKS TO INSURE THAT ALL TOURS IN AN OVERLAY SEGMENT HAVE BEEN         550
C SELECTED IN A COMMAND OR THAT THEY HAVE ALL BEEN OMITTED                551
C                                                                         552
      COMMON/PNTRS/IOVRLY,IOVTR(2),                                       553
     1NPCTDT,NPCTRD,LPCTDT,LNMLST(4),NNAMES(4),NDAYDT,LDAYNM,             554
     2LDYRFL,NDAYRD,LDYWFL,NTRDT,LTRTB(2),LTRST,LTREND,LTRRFL,LTRNM,      555
     3NTRRD,LTRWFL,NBLDT,LBLKTB(2),LBLRFL,NBLRD,LBLWFL,NDIVDT,NDIVRD,     556
     4LDIVNM,LDIVFL                                                       557
C                                                                         558
      COMMON/SYSTEM/SYSIN,SYSOUT,IFILE,LIT                                559
      INTEGER SYSIN,SYSOUT                                                560
C                                                                         561
C                                                                         562
      COMMON/STORE/TOP,BOT,RDBOT,MAXBOT,NWORDS,CDAT(6000),C2DAT(6000)     563
      INTEGER TOP,BOT,RDBOT                                               564
      DIMENSION ICDAT(6000),IC2DAT(6000)                                  565
      EQUIVALENCE(ICDAT,CDAT),(IC2DAT,C2DAT)                              566
C                                                                         567
C                                                                         568
      IERR=0                                                              569
      IOV=ICDAT(LTRRFL+NTRDT-1)                                           570
      IF(IOVRLY .EQ. 0 .OR. IOV .EQ. 0) RETURN                            571
      IOV=ICDAT(LTRWFL+IOV-1)                                             572
      IOV1=ICDAT(LTRWFL+IOVTR(1)-1)                                       573
      IOV2=ICDAT(LTRWFL+IOVTR(2)-1)                                       574
      IF(IOV+IOV1+IOV2 .EQ. 0) RETURN                                     575
      IF(IOV .NE. 0 .AND. IOV1 .NE. 0 .AND. IOV2 .NE. 0) RETURN           576
      WRITE(SYSOUT,1)                                                     577
1     FORMAT(/' *** INVALID OVERLAY TOUR SPECIFICATION - REENTER.')       578
      IERR=1                                                              579
      RETURN                                                              580
      END                                                                 581
```


SUBROUTINE COMPTB

Subroutine COMPTB (compute table) is called from the routines that control DISP command output to compute measures for one shift. Parameter ITAB specifies the table for which measures are to be computed. LPCT, LDAY, and LTOUR are pointers to the data for the precinct, day, and tour to be used in the computation. ITYPE is the position of the tour relative to all tours in the database; it provides an index to tour starting and ending times. IADD indicates whether the measures computed for the tour are to be included in the next higher level of aggregation (this depends on the DISP command output order and on whether the shift is an overlay).

For each of the output tables, weighted sums and weights are computed for all measures and summed over all blocks of a shift. The measures are either computed directly from data items in CDAT and C2DAT or by function references to such routines as OBJF1 for fraction of calls delayed. Weighted sums and weights are accumulated in the columns of row 4 of arrays T and S, respectively. If requested, the contents of row 4 of arrays T and S are added to the contents of row 3; this represents inclusion of the measures for the shift in the next higher level of aggregation. Finally, averages of the measures over the blocks of the shift are computed by dividing the weighted sums in T by the weights in S.

```
      SUBROUTINE COMPTB(ITAB,LPCT,LDAY,LTOUR,ITYPE,IADD)                  582
C                                                                         583
C  COMPUTES ONE OUTPUT LINE OF ONE TABLE                                  584
C                                                                         585
      COMMON/OFFSET/NMPOFF,DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,     586
     1NWDPCT,CPDOFF,SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,NWDDY,             587
     2QDTOFF,QXTOFF,CRTOFF,QOTOFF,QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,     588
     3PVTOFF,HFTOFF,MFTOFF,LFTOFF,NPRIO,NWDTR,BLDOFF,QOBOFF,QNBOFF,       589
     4EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,NWDBL              590
C                                                                         591
      INTEGER DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,CPDOFF,           592
     1SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,QDTOFF,QXTOFF,CRTOFF,QOTOFF,     593
     2QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,PVTOFF,HFTOFF,BLDOFF,            594
     3EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,QOBOFF,QNBOFF      595
C                                                                         596
C                                                                         597
```

```
      COMMON/STORE/TOP,BOT,RDBOT,MAXBOT,NWORDS,CDAT(6000),C2DAT(6000)
      INTEGER TOP,BOT,RDBOT
      DIMENSION ICDAT(6000),IC2DAT(6000)
      EQUIVALENCE(ICDAT,CDAT),(IC2DAT,C2DAT)
C
C
       COMMON/PNTRS/IOVRLY,IOVTR(2),
     1NPCTDT,NPCTRD,LPCTDT,LNMLST(4),NNAMES(4),NDAYDT,LDAYNM,
     2LDYRFL,NDAYRD,LDYWFL,NTRDT,LTRTB(2),LTRST,LTREND,LTRRFL,LTRNM,
     3NTRRD,LTRWFL,NBLDT,LBLKTB(2),LBLRFL,NBLRD,LBLWFL,NDIVDT,NDIVRD,
     4LDIVNM,LDIVFL
C
       COMMON/STATS/T(4,8),S(4,8),PORDER(3),RORDER(3),CIND(8)
       INTEGER PORDER,RORDER
C
       DATA BLANK/1H /,STAR/1H*/
       DIMENSION ICNSTR(10)
       DATA ICNSTR(1)/1/,ICNSTR(2)/3/,ICNSTR(3)/7/,ICNSTR(4)/4/,
     1 ICNSTR(5)/5/,ICNSTR(6)/1/,ICNSTR(7)/5/,ICNSTR(8)/6/,
     2 ICNSTR(9)/7/,ICNSTR(10)/8/
       DIMENSION C1(3),C2(3),C3(3),CPC(3)
C
       IEND=ICDAT(LTREND+ITYPE-1)
       ISTART=ICDAT(LTRST+ITYPE-1)
       ILEN=IEND-ISTART+1
       TOURLN=ILEN
       LCR=LDAY+CRDOFF
       LST=LDAY+STDOFF
C
C                                      ***  NO. CARS
       ACT=CDAT(LTOUR+ACTOFF)
       T(4,1)=ACT
       S(4,1)=1.
C                                      ***  CAR HOURS
       T(4,2)=ACT*TOURLN
       S(4,2)=1.
C
C
       GO TO (100,200,300,400,500),ITAB
C
C ***************************************************************
C *   COMPUTE: TABLE 1. WORKLOADS OF PATROL CARS          *
C ***************************************************************
C
100    CONTINUE
C
       DO 120 IBLK=1,2
       IBLD=ICDAT(LTRTB(IBLK)+ITYPE-1)
       IF(IBLD .LT. 1) GO TO 120
       IBLR=ICDAT(LBLRFL+IBLD-1)
       LBLK=LDAY+BLDOFF+(IBLR-1)*NWDBL
       ISTART=ICDAT(LBLKTB(1)+IBLD-1)
```



```
       IEND=ICDAT(LBLKTB(2)+IBLD-1)
       BLKLN=IEND-ISTART+1
       EF=CDAT(LBLK+EFBOFF)
C
C          IF OVERLAY TOUR, GET DATA FROM OVERLAID TOUR
C
       LTTOUR=LTOUR
       IF(ICDAT(LTOUR+TYTOFF) .EQ. 5)
     1      LTTOUR=LDAY+TRDOFF+(IOVTR(IBLK)-1)*NWDTR
       LFR=LTTOUR+HFTOFF
C
       RV=CDAT(LTTOUR+RVTOFF)
       AWL=CDAT(LBLK+AWBOFF)
C
       DO 101 JC=1,3
       C1(JC)=C2DAT(LBLK+QDTOFF+JC-1)
       C2(JC)=C2DAT(LBLK+QOTOFF+JC-1)
101    C3(JC)=C2DAT(LBLK+TYTOFF+JC-1)
C
C                                      ***  CALL RATE
C
       T(4,3)=T(4,3)+CDAT(LBLK+CRBOFF)
       S(4,3)=S(4,3)+BLKLN
C
C                                      ***  SERVICE TIME
C
       FMLCAR=CDAT(LFR)  *(C1(1)+2.*C1(2)+3.*C1(3))   +
     1        CDAT(LFR+1)*(C2(1)+2.*C2(2)+3.*C2(3))   +
     1        CDAT(LFR+2)*(C3(1)+2.*C3(2)+3.*C3(3))
       ST=CDAT(LBLK+AWBOFF)*BLKLN*60./FMLCAR
C
       T(4,4)=T(4,4)+ST
       S(4,4)=S(4,4)+CDAT(LBLK+CRBOFF)
C
C                                      ***  PERCENT TIME BUSY (CFS)
C
       ACT=CDAT(LBLK+ACBOFF)
       X=AWL/ACT
       Y=BLKLN*ACT
       T(4,5)=T(4,5)+X*Y*100.0
       S(4,5)=S(4,5)+Y
C
C                                      ***  PERCENT TIME BUSY (NONCFS)
C
       X=1.0-(EF/ACT)
       Y=BLKLN*ACT
       T(4,6)=T(4,6)+X*Y*100.0
       S(4,6)=S(4,5)
C
C                                      ***  PERCENT TIME BUSY (TOTAL)
C
       T(4,7)=T(4,5)+T(4,6)
```

```
      S(4,7)=S(4,5)                                                      702
C                                                                        703
C                                  ***  AVG. CARS AVAILABLE              704
C                                                                        705
      X=EF-AWL                                                           706
C                                                                        707
      IF(X .LT. 0.0) X=0.0                                               708
C                                                                        709
      T(4,8)=T(4,8)+X*BLKLN                                              710
      S(4,8)=S(4,8)+BLKLN                                                711
120   CONTINUE                                                           712
C                                                                        713
C          ACCUMULATE MEASURES IF REQUESTED                              714
C                                                                        715
      N=8                                                                716
      IF(IADD .LT. 1) N=2                                                717
      DO 130 I=1,N                                                       718
      IF(S(4,I) .EQ. 0.) T(4,I)=0.                                       719
      T(3,I)=T(3,I)+T(4,I)                                               720
      S(3,I)=S(3,I)+S(4,I)                                               721
130   CONTINUE                                                           722
C                                                                        723
C           COMPUTE AVERAGES                                             724
C                                                                        725
140   DO 150 I=1,8                                                       726
      CIND(I)=BLANK                                                      727
      IF(S(4,I) .EQ. 0.) GO TO 150                                       728
      T(4,I)=T(4,I)/S(4,I)                                               729
150   CONTINUE                                                           730
C                                                                        731
      RETURN                                                             732
C                                                                        733
C ***********************************************************************734
C *          END OF TABLE 1 COMPUTATIONS                              *  735
C ***********************************************************************736
C                                                                        737
C ***********************************************************************738
C *   COMPUTE: TABLE 2. TIME ALLOCATION: CARS STARTING THE TOUR *        739
C ***********************************************************************740
C                                                                        741
C                                                                        742
200   CONTINUE                                                           743
C                                                                        744
      DO 220 IBLK=1,2                                                    745
      IBLD=ICDAT(LTRTB(IBLK)+ITYPE-1)                                    746
      IF(IBLD .LT. 1) GO TO 220                                          747
      IBLR=ICDAT(LBLRFL+IBLD-1)                                          748
      LBLK=LDAY+BLDOFF+(IBLR-1)*NWDBL                                    749
      ISTART=ICDAT(LBLKTB(1)+IBLD-1)                                     750
      IEND=ICDAT(LBLKTB(2)+IBLD-1)                                       751
      BLKLN=IEND-ISTART+1                                                752
      EF=CDAT(LBLK+EFBOFF)                                               753
```



```
C                                                                        754
C                                                                        755
C          IF OVERLAY TOUR, GET DATA FROM OVERLAID TOUR                  756
      LTTOUR=LTOUR                                                       757
      IF(ICDAT(LTOUR+TYTOFF) .EQ. 5)                                     758
     1      LTTOUR=LDAY+TRDOFF+(IOVTR(IBLK)-1)*NWDTR                    759
      LFR=LTTOUR+HFTOFF                                                  760
C                                                                        761
      RV=CDAT(LTTOUR+RVTOFF)                                             762
      AWL=CDAT(LBLK+AWBOFF)                                              763
C                                                                        764
C                                  ***  OFFICER HOURS                   765
C                                                                        766
      PCTOFF=C2DAT(LTOUR+QDTOFF+9)                                       767
      T(4,3)=T(4,3)+((T(4,2)*(1.0+(PCTOFF/100.)))*BLKLN)                 768
      S(4,3)=S(4,3)+BLKLN                                                769
C                                                                        770
C                                  ***  UNCOMM HRS/CAR                  771
C                                                                        772
      ACTB=CDAT(LBLK+ACBOFF)                                             773
      X=(EF-AWL)*BLKLN/ACTB                                              774
       Y=T(4,1)                                                          775
      T(4,4)=T(4,4)+X*Y                                                  776
      S(4,4)=Y                                                           777
C                                                                        778
C                                  ***  % TIME UNCOMM                   779
C                                                                        780
      X=(EF-AWL)/ACTB                                                    781
      Y=BLKLN*T(4,1)                                                     782
      T(4,5)=T(4,5)+X*Y*100.0                                            783
      S(4,5)=S(4,5)+Y                                                    784
C                                                                        785
220   CONTINUE                                                           786
C                                                                        787
C                                                                        788
C          ACCUMULATE MEASURES IF REQUESTED                              789
C                                                                        790
      N=5                                                                791
      DO 230 I=1,N                                                       792
      IF(S(4,I) .EQ. 0.) T(4,I)=0.                                       793
      T(3,I)=T(3,I)+T(4,I)                                               794
      S(3,I)=S(3,I)+S(4,I)                                               795
230   CONTINUE                                                           796
C                                                                        797
C                                                                        798
C          COMPUTE AVERAGES                                              799
C                                                                        800
240   DO 250 I=1,5                                                       801
      CIND(I)=BLANK                                                      802
      IF(S(4,I) .EQ. 0.) GO TO 250                                       803
      T(4,I)=T(4,I)/S(4,I)                                               804
250   CONTINUE                                                           805
C
```

```
      RETURN                                                            806
C                                                                       807
C ***********************************************************************808
C *          END OF TABLE 2 COMPUTATIONS                              * 809
C ***********************************************************************810
C                                                                       811
C ***********************************************************************812
C *   COMPUTE: TABLE 3. AVERAGE DELAYS OF CALLS                       * 813
C ***********************************************************************814
C                                                                       815
300   CONTINUE                                                          816
C                                                                       817
      DO 320 IBLK=1,2                                                   818
      IBLD=ICDAT(LTRTB(IBLK)+ITYPE-1)                                   819
      IF(IBLD .LT. 1) GO TO 320                                         820
      IBLR=ICDAT(LBLRFL+IBLD-1)                                         821
      LBLK=LDAY+BLDOFF+(IBLR-1)*NWDBL                                   822
      ISTART=ICDAT(LBLKTB(1)+IBLD-1)                                    823
      IEND=ICDAT(LBLKTB(2)+IBLD-1)                                      824
      BLKLN=IEND-ISTART+1                                               825
      EF=CDAT(LBLK+EFBOFF)                                              826
C                                                                       827
C          IF OVERLAY TOUR, GET DATA FROM OVERLAID TOUR                 828
C                                                                       829
      LTTOUR=LTOUR                                                      830
      IF(ICDAT(LTOUR+TYTOFF) .EQ. 5)                                    831
     1      LTTOUR=LDAY+TRDOFF+(IOVTR(IBLK)-1)*NWDTR                    832
      LFR=LTTOUR+HFTOFF                                                 833
C                                                                       834
      RV=CDAT(LTTOUR+RVTOFF)                                            835
      AWL=CDAT(LBLK+AWBOFF)                                             836
C                                                                       837
      DO 301 JC=1,3                                                     838
      C1(JC)=C2DAT(LBLK+QDTOFF+JC-1)                                    839
      C2(JC)=C2DAT(LBLK+QOTOFF+JC-1)                                    840
301   C3(JC)=C2DAT(LBLK+TYTOFF+JC-1)                                    841
C                                                                       842
C                                                                       843
C                          ***  PRTY 2 QUEUE                          844
C                                                                       845
      X=OBJF2(2,ISTART,IEND,LPCT,LCR,LST,LFR,RV,EF,C1,C2,C3)            846
      Y=CDAT(LBLK+CRBOFF)*CDAT(LFR+1)                                   847
      T(4,3)=T(4,3)+X                                                   848
      S(4,3)=S(4,3)+Y                                                   849
C                                                                       850
C                          ***  PRTY 2 DELAYS + TRAVEL                 851
C                                                                       852
      X=OBJF3(2,ISTART,IEND,LPCT,LCR,LST,LFR,RV,EF,C1,C2,C3)            853
      Y=CDAT(LBLK+CRBOFF)*CDAT(LFR+1)                                   854
      T(4,4)=T(4,4)+X                                                   855
      S(4,4)=S(4,4)+Y                                                   856
C                                                                       857
C                          ***  PRTY 3 QUEUE
```



```
C                                                                       858
      X=OBJF2(3,ISTART,IEND,LPCT,LCR,LST,LFR,RV,EF,C1,C2,C3)            859
      Y=CDAT(LBLK+CRBOFF)*CDAT(LFR+2)                                   860
      T(4,5)=T(4,5)+X                                                   861
      S(4,5)=S(4,5)+Y                                                   862
C                                                                       863
C                                                                       864
C                          ***  PRTY 3 DELAYS + TRAVEL                 865
      X=OBJF3(3,ISTART,IEND,LPCT,LCR,LST,LFR,RV,EF,C1,C2,C3)            866
      Y=CDAT(LBLK+CRBOFF)*CDAT(LFR+2)                                   867
      T(4,6)=T(4,6)+X                                                   868
      S(4,6)=S(4,6)+Y                                                   869
C                                                                       870
C                                                                       871
C                          ***  AVERAGE QUEUE                         872
      X=OBJF2(0,ISTART,IEND,LPCT,LCR,LST,LFR,RV,EF,C1,C2,C3)            873
      Y=CDAT(LBLK+CRBOFF)                                               874
      T(4,7)=T(4,7)+X                                                   875
      S(4,7)=S(4,7)+Y                                                   876
C                                                                       877
C                                                                       878
C                          ***  AVERAGE DELAY + TRAVEL                 879
      X=OBJF3(0,ISTART,IEND,LPCT,LCR,LST,LFR,RV,EF,C1,C2,C3)            880
      Y=CDAT(LBLK+CRBOFF)                                               881
      T(4,8)=T(4,8)+X                                                   882
      S(4,8)=S(4,8)+Y                                                   883
320   CONTINUE                                                          884
C                                                                       885
C          ACCUMULATE MEASURES IF REQUESTED                             886
C                                                                       887
      N=8                                                               888
      IF(IADD .LT. 1) N=2                                               889
      DO 330 I=1,N                                                      890
      IF(S(4,I) .EQ. 0.) T(4,I)=0.                                      891
      T(3,I)=T(3,I)+T(4,I)                                              892
      S(3,I)=S(3,I)+S(4,I)                                              893
330   CONTINUE                                                          894
C                                                                       895
C          COMPUTE AVERAGES                                             896
C                                                                       897
340   DO 350 I=1,8                                                      898
      CIND(I)=BLANK                                                     899
      IF(S(4,I) .EQ. 0.) GO TO 350                                      900
      T(4,I)=T(4,I)/S(4,I)                                              901
350   CONTINUE                                                          902
C                                                                       903
      RETURN                                                            904
C                                                                       905
C ***********************************************************************906
C *          END OF TABLE 3 COMPUTATIONS                              * 907
C ***********************************************************************908
                                                                        909
```

```
C
C ***************************************************************
C *   COMPUTE: TABLE 4. CALLS DELAYED AND PATROL INTERVAL *
C ***************************************************************
C
400     CONTINUE
C
        DO 420 IBLK=1,2
        IBLD=ICDAT(LTRTB(IBLK)+ITYPE-1)
        IF(IBLD .LT. 1) GO TO 420
        IBLR=ICDAT(LBLRFL+IBLD-1)
        LBLK=LDAY+BLDOFF+(IBLR-1)*NWDBL
        ISTART=ICDAT(LBLKTB(1)+IBLD-1)
        IEND=ICDAT(LBLKTB(2)+IBLD-1)
        BLKLN=IEND-ISTART+1
        EF=CDAT(LBLK+EFBOFF)
C
C           IF OVERLAY TOUR, GET DATA FROM OVERLAID TOUR
C
        LTTOUR=LTOUR
        IF(ICDAT(LTOUR+TYTOFF) .EQ. 5)
     1      LTTOUR=LDAY+TRDOFF+(IOVTR(IBLK)-1)*NWDTR
        LFR=LTTOUR+HFTOFF
C
        RV=CDAT(LTTOUR+RVTOFF)
        AWL=CDAT(LBLK+AWBOFF)
C
        DO 401 JC=1,3
        C1(JC)=C2DAT(LBLK+QDTOFF+JC-1)
        C2(JC)=C2DAT(LBLK+QOTOFF+JC-1)
401     C3(JC)=C2DAT(LBLK+TYTOFF+JC-1)
C
        X=OBJF1(0,ISTART,IEND,LPCT,LCR,LST,LFR,RV,EF,C1,C2,C3)
C
C                                   ***  % CALLS DELAYED PRTY1
C
        Y=CDAT(LBLK+CRBOFF)
        T(4,3)=T(4,3)+X*100.0
        S(4,3)=S(4,3)+Y
C
C                                   ***  % CALLS DELAYED PRTY2
C
        Y=CDAT(LBLK+CRBOFF)
        T(4,4)=T(4,4)+X*100.0
        S(4,4)=S(4,4)+Y
C
C                                   ***  % CALLS DELAYED PRTY3
C
        Y=CDAT(LBLK+CRBOFF)
        T(4,5)=T(4,5)+X*100.0
        S(4,5)=S(4,5)+Y
C
```



```
C
C                                   ***  % CALLS DELAYED TOTAL
        Y=CDAT(LBLK+CRBOFF)
        T(4,6)=T(4,6)+X*100.0
        S(4,6)=S(4,6)+Y
C
C
C                                   ***  PATROL INTERVAL
        SPEED=CDAT(LTTOUR+PVTOFF)
        STMILE=CDAT(LPCT+SMPOFF)
        X=STMILE/(SPEED*(EF-AWL))*BLKLN
        T(4,7)=T(4,7)+X
        S(4,7)=S(4,7)+BLKLN
C
420     CONTINUE
C
C           ACCUMULATE MEASURES IF REQUESTED
C
        N=7
        IF(IADD .LT. 1) N=2
        DO 430 I=1,N
        IF(S(4,I) .EQ. 0.) T(4,I)=0.
        T(3,I)=T(3,I)+T(4,I)
        S(3,I)=S(3,I)+S(4,I)
430     CONTINUE
C
C           COMPUTE AVERAGES
C
440     DO 450 I=1,7
        CIND(I)=BLANK
        IF(S(4,I) .EQ. 0.) GO TO 450
        T(4,I)=T(4,I)/S(4,I)
450     CONTINUE
C
        RETURN
C
C
C ***************************************************************
C *           END OF TABLE 4 COMPUTATIONS                       *
C ***************************************************************
C
C ***************************************************************
C *   COMPUTE: TABLE 5. PRIORITY 1 DELAYS                       *
C ***************************************************************
C
500     CONTINUE
C
        DO 520 IBLK=1,2
        IBLD=ICDAT(LTRTB(IBLK)+ITYPE-1)
        IF(IBLD .LT. 1) GO TO 520
        IBLR=ICDAT(LBLRFL+IBLD-1)
        LBLK=LDAY+BLDOFF+(IBLR-1)*NWDBL
```

```
      ISTART=ICDAT(LBLKTB(1)+IBLD-1)                                    1014
      IEND=ICDAT(LBLKTB(2)+IBLD-1)                                      1015
      BLKLN=IEND-ISTART+1                                               1016
      EF=CDAT(LBLK+EFBOFF)                                              1017
C                                                                       1018
C          IF OVERLAY TOUR, GET DATA FROM OVERLAID TOUR                 1019
C                                                                       1020
      LTTOUR=LTOUR                                                      1021
      IF(ICDAT(LTOUR+TYTOFF) .EQ. 5)                                    1022
     1     LTTOUR=LDAY+TRDOFF+(IOVTR(IBLK)-1)*NWDTR                     1023
      LFR=LTTOUR+HFTOFF                                                 1024
C                                                                       1025
      RV=CDAT(LTTOUR+RVTOFF)                                            1026
      AWL=CDAT(LBLK+AWBOFF)                                             1027
C                                                                       1028
      DO 501 JC=1,3                                                     1029
      C1(JC)=C2DAT(LBLK+QDTOFF+JC-1)                                    1030
      C2(JC)=C2DAT(LBLK+QOTOFF+JC-1)                                    1031
501   C3(JC)=C2DAT(LBLK+TYTOFF+JC-1)                                    1032
C                                                                       1033
      PHP1=CDAT(LFR)                                                    1034
      PHP2=CDAT(LFR+1)                                                  1035
      PHP3=(1.0-PHP1-PHP2)                                              1036
C                                                                       1037
      DO 10 I=1,3                                                       1038
      CPC(I)=0.0                                                        1039
10    CONTINUE                                                          1040
C                                                                       1041
      DO 20 I=1,3                                                       1042
      CPC(1)=CPC(1)+(I*C1(I))                                           1043
      CPC(2)=CPC(2)+(I*C2(I))                                           1044
      CPC(3)=CPC(3)+(I*C3(I))                                           1045
20    CONTINUE                                                          1046
C                                                                       1047
      AVGCPC=(PHP1*CPC(1))+(PHP2*CPC(2))+(PHP3*CPC(3))                  1048
C                                                                       1049
C                                                                       1050
C                                                                       1051
C                              ***  AVG CARS/CFS  PRTY1                 1052
      T(4,3)=T(4,3)+CPC(1)*BLKLN                                        1053
      S(4,3)=S(4,3)+BLKLN                                               1054
C                                                                       1055
C                                                                       1056
C                              ***  AVG CARS/CFS PRTY2                  1057
      T(4,4)=T(4,4)+CPC(2)*BLKLN                                        1058
      S(4,4)=S(4,4)+BLKLN                                               1059
C                                                                       1060
C                                                                       1061
C                              ***  AVG CARS/CFS PRTY3                  1062
      T(4,5)=T(4,5)+CPC(3)*BLKLN                                        1063
      S(4,5)=S(4,5)+BLKLN                                               1064
                                                                        1065
```



```
C                                                                       1066
C                              ***  AVG CARS/CFS TOTAL                  1067
C                                                                       1068
      T(4,6)=T(4,6)+AVGCPC*BLKLN                                        1069
      S(4,6)=S(4,6)+BLKLN                                               1070
C                                                                       1071
C                              ***  PRTY 1 QUEUE                        1072
C                                                                       1073
      X=OBJF2(1,ISTART,IEND,LPCT,LCR,LST,LFR,RV,EF,C1,C2,C3)            1074
      Y=CDAT(LBLK+CRBOFF)*CDAT(LFR)                                     1075
      T(4,7)=T(4,7)+X                                                   1076
      S(4,7)=S(4,7)+Y                                                   1077
C                                                                       1078
C                              ***  PRTY 1 DELAYS + TRAVEL              1079
C                                                                       1080
      X=OBJF3(1,ISTART,IEND,LPCT,LCR,LST,LFR,RV,EF,C1,C2,C3)            1081
      Y=CDAT(LBLK+CRBOFF)*CDAT(LFR)                                     1082
      T(4,8)=T(4,8)+X                                                   1083
      S(4,8)=S(4,8)+Y                                                   1084
C                                                                       1085
520   CONTINUE                                                          1086
C                                                                       1087
C          ACCUMULATE MEASURES IF REQUESTED                             1088
C                                                                       1089
      N=8                                                               1090
      IF(IADD .LT. 1) N=2                                               1091
      DO 530 I=1,N                                                      1092
      IF(S(4,I) .EQ. 0.) T(4,I)=0.                                      1093
      T(3,I)=T(3,I)+T(4,I)                                              1094
      S(3,I)=S(3,I)+S(4,I)                                              1095
530   CONTINUE                                                          1096
C                                                                       1097
C          COMPUTE AVERAGES                                             1098
C                                                                       1099
      DO 550 I=1,8                                                      1100
540   CIND(I)=BLANK                                                     1101
      IF(S(4,I) .EQ. 0.) GO TO 550                                      1102
      T(4,I)=T(4,I)/S(4,I)                                              1103
550   CONTINUE                                                          1104
C                                                                       1105
      RETURN                                                            1106
C                                                                       1107
      END
```

FUNCTION CRLEFT

Function CRLEFT is called when the user chooses the option of smoothing queuing behavior over time. It returns an estimate of the number of calls waiting in the queue at the beginning of a block, based on the expected waiting time of calls received during the last hour of the previous block.

The algorithm used to carry over work between blocks is described in App. A of the User's Manual.

```
      FUNCTION CRLEFT(CRM1,STM1,PDM1,EFM1,PHP1,PHP2,C1,C2,C3)          1108
C                                                                      1109
C     ESTIMATE CALLS WAITING IN QUEUE AT BEGINNING OF BLOCK            1110
C                                                                      1111
      DIMENSION C1(3),C2(3),C3(3)                                      1112
C                                                                      1113
C                                                                      1114
      WM1=TRIDSP(2,0,CRM1,STM1,PHP1,PHP2,EFM1,C1,C2,C3,PI)/60.0        1115
      QM1A=CRM1*WM1                                                    1116
      PAVL = 1.0 - PDM1                                                1117
      IF (PAVL .LE. .001) GOTO 20                                      1118
      CONDW=WM1/PAVL                                                   1119
      IF (CONDW .GE. 1.0) GO TO 20                                     1120
   15 QM1B=PDM1*CRM1*CONDW                                             1121
      GO TO 30                                                         1122
   20 QM1B=PDM1*CRM1                                                   1123
   30 CRLEFT=MIN(QM1A,QM1B)                                            1124
         QM1AB=CRLEFT-(EFM1-STM1)                                      1125
         IF(QM1AB .GT. 0.0) CRLEFT=(CRLEFT+QM1AB)/2.0                  1126
C                                                                      1127
      RETURN                                                           1128
      END                                                              1129
```


SUBROUTINE DERIVE

Subroutine DERIVE is called by READ and SET after reading or modifying the data for a day for a precinct. The two parameters LPCT and LDAY are pointers to the data for the precinct and the day, respectively. IREAD controls the printing or suppression of output messages. DERIVE's primary function is to determine, for each block of the day: number of actual cars on duty, average cfs workload, maximum cfs workload in any hour, total calls for service, and number of effective cars on duty. For each tour of the day, DERIVE determines the fraction of calls in the lowest priority class and the total number of calls during the tour. DERIVE also initializes the vectors of effective cars and delay probabilities that are needed by the smoothing algorithm, if the queuing behavior is to be smoothed over time.

Subroutine SBLACT (*set block actual cars*) is called to determine the number of actual cars on duty in each block of a day from the number of cars assigned to each tour of the day. The number of effective cars on duty in a block is computed using the formula given in Sec. III.

DERIVE also checks to determine whether each block has enough effective cars to handle the cfs workload in its busiest hour. (If smoothing is being used, the check is for enough effective cars to handle the average cfs workload in the block or the workload in the last hour of the block.) If a block lacks sufficient effective cars, then the algorithm described in Sec. III is used to determine where to increase the assignment of cars to tours of a day so that each block will have enough effective cars. After this algorithm has been applied, subroutine SBLEF (*set block effective cars*) is called to redetermine the number of effective cars in each block of the day.

```
      SUBROUTINE DERIVE(LPCT,LDAY,IREAD)                               1130
C                                                                      1131
C  CALCULATES, FOR EACH BLOCK IN DAY AND PRECINCT,                     1132
C  AVERAGES OF INPUT DATA                                              1133
C                                                                      1134
      COMMON/KEYWDS/NKYWD,NTYPES,TYPOFF(4),KEYWD(8,30),WDTYPE(30)      1135
      INTEGER TYPOFF,WDTYPE                                            1136
      DIMENSION PCLSNM(8),DCLSNM(8),TOURNM(8)                          1137
      EQUIVALENCE (PCLSNM,KEYWD(1,4)),(DCLSNM,KEYWD(1,3)),             1138
     1(TOURNM,KEYWD(1,2))                                              1139
```

```
C                                                                       1140
      COMMON/SYSTEM/SYSIN,SYSOUT,IFILE,LIT                              1141
      INTEGER SYSIN,SYSOUT                                              1142
C                                                                       1143
     COMMON/STORE/TOP,BOT,RDBOT,MAXBOT,NWORDS,CDAT(6000),C2DAT(6000)    1144
     INTEGER TOP,BOT,RDBOT                                              1145
     DIMENSION ICDAT(6000),IC2DAT(6000)                                 1146
     EQUIVALENCE(ICDAT,CDAT),(IC2DAT,C2DAT)                             1147
C                                                                       1148
      COMMON/OFFSET/NMPOFF,DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,   1149
    1NWDPCT,CPDOFF,SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,NWDDY,            1150
    2QDTOFF,QXTOFF,CRTOFF,QOTOFF,QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,    1151
    3PVTOFF,HFTOFF,MFTOFF,LFTOFF,NPRIO,NWDTR,BLDOFF,QOBOFF,QNBOFF,      1152
    4EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,NWDBL             1153
C                                                                       1154
      INTEGER DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,CPDOFF,         1155
    1SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,QDTOFF,QXTOFF,CRTOFF,QOTOFF,    1156
    2QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,PVTOFF,HFTOFF,BLDOFF,           1157
    3EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,QOBOFF,QNBOFF     1158
C                                                                       1159
      COMMON/PNTRS/IOVRLY,IOVTR(2),                                     1160
    1NPCTDT,NPCTRD,LPCTDT,LNMLST(4),NNAMES(4),NDAYDT,LDAYNM,            1161
    2LDYRFL,NDAYRD,LDYWFL,NTRDT,LTRTB(2),LTRST,LTREND,LTRRFL,LTRNM,     1162
    3NTRRD,LTRWFL,NBLDT,LBLKTB(2),LBLRFL,NBLRD,LBLWFL,NDIVDT,NDIVRD,    1163
    4LDIVNM,LDIVFL                                                      1164
C                                                                       1165
      REAL LFR                                                          1166
      DIMENSION IBERR(24),ACB(2),ACTR(2)                                1167
      DIMENSION C1(3),C2(3),C3(3)                                       1168
C                                                                       1169
       ADDSUM=0.0                                                       1170
C                                                                       1171
      IDERR=0                                                           1172
       ISMFLG = IC2DAT(1)                                               1173
C                                                                       1174
      B1=CDAT(LPCT+B1POFF)                                              1175
      B2=CDAT(LPCT+B2POFF)                                              1176
C                                                                       1177
C          FIND NUMBER OF CARS ON DUTY IN EACH BLOCK                    1178
C                                                                       1179
      CALL SBLACT(LPCT,LDAY)                                            1180
C                                                                       1181
       LST=LDAY+STDOFF                                                  1182
       LCR=LDAY+CRDOFF                                                  1183
C                                                                       1184
      DO 20 IBLDT=1,NBLDT                                               1185
C                                                                       1186
       IF(IBLDT .NE. 1 .OR .ISMFLG .EQ. 0) GO TO 21                     1187
       C2DAT(LST)=-1.0                                                  1188
       C2DAT(LCR)=0.0                                                   1189
C                                                                       1190
21    IBLK=ICDAT(LBLRFL+IBLDT-1)                                        1191
```



```
      IF(IBLK .EQ. 0) GO TO 20                                          1192
      LBLK=LDAY+BLDOFF+(IBLK-1)*NWDBL                                   1193
      IBERR(IBLK)=0                                                     1194
      ISTART=ICDAT(LBLKTB(1)+IBLDT-1)                                   1195
      IEND=ICDAT(LBLKTB(2)+IBLDT-1)                                     1196
      BLKLN=IEND-ISTART+1                                               1197
C                                                                       1198
       PHP2=0.0                                                         1199
       PHP3=0.0                                                         1200
       DO 23 IT=1,NTRDT                                                 1201
       ITOUR=ICDAT(LTRRFL+IT-1)                                         1202
       IF(ITOUR .LT. 1) GO TO 23                                        1203
       LTOUR=LDAY+TRDOFF+(ITOUR-1)*NWDTR                                1204
       IF(ICDAT(LTOUR+TYTOFF) .EQ. 1) GO TO 23                          1205
       ITYPE=ICDAT(LTRWFL+ITOUR-1)                                      1206
       IBL1=ICDAT(LTRTB(1)+ITYPE-1)                                     1207
       IBL2=ICDAT(LTRTB(2)+ITYPE-1)                                     1208
       IBL1=ICDAT(LBLRFL+IBL1-1)                                        1209
       IF(IBL2 .NE. 0) IBL2=ICDAT(LBLRFL+IBL2-1)                        1210
       LBLK1=LDAY+BLDOFF+(IBL1-1)*NWDBL                                 1211
       LBLK2=0                                                          1212
       IF(IBL2 .NE. 0) LBLK2=LDAY+BLDOFF+(IBL2-1)*NWDBL                 1213
       IF(LBLK1 .NE. LBLK .AND. LBLK2 .NE. LBLK) GO TO 23               1214
       PHP1=CDAT(LTOUR+HFTOFF)                                          1215
       PHP2=CDAT(LTOUR+MFTOFF)                                          1216
       PHP3=1.0-PHP1-PHP2                                               1217
       GO TO 26                                                         1218
23     CONTINUE                                                         1219
C                                                                       1220
26     CONTINUE                                                         1221
C                                                                       1222
       DO 22 JC=1,3                                                     1223
       C1(JC)=C2DAT(LBLK+QDTOFF+JC-1)                                   1224
       C2(JC)=C2DAT(LBLK+QOTOFF+JC-1)                                   1225
22     C3(JC)=C2DAT(LBLK+TYTOFF+JC-1)                                   1226
C                                                                       1227
        FMLCAR=PHP1*(C1(1)+2.*C1(2)+3.*C1(3))  +                        1228
     1         PHP2*(C2(1)+2.*C2(2)+3.*C2(3))  +                        1229
     1 (1.-PHP1-PHP2)*(C3(1)+2.*C3(2)+3.*C3(3))                         1230
C          CALCULATE AVERAGE AND MAXIMUM CALL RATE IN BLOCK             1231
C                                                                       1232
       RMAX=0.                                                          1233
       CRATE=0.                                                         1234
       AWL=0.                                                           1235
       LB=LDAY-1                                                        1236
       DO 10 I=ISTART,IEND                                              1237
       IB=I+LB                                                          1238
       CR=CDAT(IB+CRDOFF)                                               1239
       CRATE=CRATE+CR                                                   1240
         R=CR*FMLCAR*CDAT(IB+STDOFF)                                    1241
       IF(I .EQ. IEND) RLAST=R                                          1242
       IF(R .GT. RMAX) RMAX=R                                           1243
```

```
10      AWL=AWL+R                                                          1244
C                                                                          1245
        AWL=AWL/BLKLN                                                      1246
        CDAT(LBLK+CRBOFF)=CRATE                                            1247
        CDAT(LBLK+AWBOFF)=AWL                                              1248
        ACT=CDAT(LBLK+ACBOFF)                                              1249
C                                                                          1250
C            CALCULATE EFFECTIVE CARS AND CHECK WHETHER                    1251
C            MINIMUM ALLOCATION IS ACHIEVED                                1252
C                                                                          1253
        EF=ACT*(1.-((B1*AWL/ACT)+B2))                                      1254
        RMAXP=RMAX+0.0001                                                  1255
C                                                                          1256
         FFXX = RMAXP                                                      1257
C                                                                          1258
         IF ( ISMFLG .EQ. 1 ) FFXX = MAX((AWL+0.0001),RLAST)               1259
C                                                                          1260
        NEF=EF                                                             1261
        IF(FMLCAR .LT. 1.01 .AND. NEF .GT. FFXX ) GO TO 15                 1262
        IF(NEF.GT.(FFXX+.15)) GO TO 15                                     1263
        IBERR(IBLK)=1                                                      1264
        ACT=CEIL((EF+B1*AWL)/(1.-B2))                                      1265
12      EF=ACT*(1.-((B1*AWL/ACT)+B2))                                      1266
        NEF=EF                                                             1267
        IF(FMLCAR .LT. 1.01 .AND. NEF .GT. FFXX ) GO TO 13                 1268
        IF(NEF.GT.(FFXX+.15)) GO TO 13                                     1269
        ACT=ACT+1.                                                         1270
        GO TO 12                                                           1271
13      CDAT(LBLK+ACBOFF)=ACT                                              1272
15      CDAT(LBLK+EFBOFF)=EF                                               1273
        CDAT(LBLK+RMBOFF)=RMAX                                             1274
C                                                                          1275
         IF(ISMFLG .EQ. 0) GO TO 20                                        1276
         DO 17 I=ISTART,IEND                                               1277
         I1=I+1                                                            1278
         IF(I1 .GT. 24) GO TO 20                                           1279
         C2DAT(LST+I)=EF                                                   1280
         CRM1=CDAT(LCR+I)                                                  1281
         STM1=CDAT(LST+I)                                                  1282
         C2DAT(LCR+I)=TRIDSP(1,0,CRM1,STM1,PHP1,PHP2,EF,C1,C2,C3,PI)       1283
17       CONTINUE                                                          1284
C                                                                          1285
20      CONTINUE                                                           1286
C                                                                          1287
        DO 100 ITYPE=1,NTRDT                                               1288
        ITOUR=ICDAT(LTRRFL+ITYPE-1)                                        1289
        IF(ITOUR .LT. 1) GO TO 100                                         1290
        ITERR=0                                                            1291
        LTOUR=LDAY+TRDOFF+(ITOUR-1)*NWDTR                                  1292
        IF(ICDAT(LTOUR+TYTOFF) .EQ. 1) GO TO 100                           1293
C                                                                          1294
C            CALL RATE IN TOURS                                            1295
```



```
C                                                                          1296
         IF(NPRIO .LT. 2) GO TO 40                                         1297
         LFR=1.                                                            1298
         N=NPRIO-1                                                         1299
         DO 30 I=1,N                                                       1300
30       LFR=LFR-CDAT(LTOUR+HFTOFF+I-1)                                    1301
         CDAT(LTOUR+HFTOFF+N)=LFR                                          1302
40       CRATE=0.                                                          1303
         DO 50 IBLK=1,2                                                    1304
         ACB(IBLK)=0.                                                      1305
         IBLD=ICDAT(LTRTB(IBLK)+ITYPE-1)                                   1306
         IF(IBLD .LT. 1) GO TO 50                                          1307
         IBLR=ICDAT(LBLRFL+IBLD-1)                                         1308
         LBLK=LDAY+BLDOFF+(IBLR-1)*NWDBL                                   1309
         ACB(IBLK)=CDAT(LBLK+ACBOFF)                                       1310
         CRATE=CRATE+CDAT(LBLK+CRBOFF)                                     1311
         IF(IBERR(IBLR) .EQ. 0) GO TO 50                                   1312
         ITERR=ITERR+IBLK                                                  1313
50       CONTINUE                                                          1314
C                                                                          1315
        CDAT(LTOUR+CRTOFF)=CRATE                                           1316
C                                                                          1317
C            CALCULATE NEW NUMBER OF ACTUAL CARS IN TOURS,                 1318
C            IF THERE HAS BEEN A CHANGE IN THE BLOCKS                      1319
         ID=ICDAT(LTOUR+TYTOFF)-1                                          1320
         GO TO (60,65,70,75),ID                                            1321
60       IF(ITERR .EQ. 0) GO TO 90                                         1322
         ACT=AMAX1(ACB(1),ACB(2))                                          1323
         GO TO 85                                                          1324
65       ACTR(1)=ACB(1)                                                    1325
         IF(ITERR .EQ. 2 .OR. ITERR .EQ. 0) GO TO 90                       1326
         ACT=ACB(1)                                                        1327
         GO TO 85                                                          1328
70       ACTR(2)=ACB(2)                                                    1329
         IF(ITERR .EQ. 1 .OR. ITERR .EQ. 0) GO TO 90                       1330
         ACT=ACB(2)                                                        1331
         GO TO 85                                                          1332
75       IF(ITERR .EQ. 0) GO TO 90                                         1333
         ACT=CDAT(LTOUR+ACTOFF)                                            1334
         TACT=ACT                                                          1335
         DO 80 I=1,2                                                       1336
80       ACT=AMAX1(ACT,ACB(I)-ACTR(I))                                     1337
         IF(TACT .GE. ACT) GO TO 90                                        1338
85       LPNM=LPCT+NMPOFF-1                                                1339
         LTNM=LTRNM+(ITYPE-1)*8-1                                          1340
         IDAY=(LDAY-DYPOFF-LPCT)/NWDDY                                     1341
         IDAY=ICDAT(LDYWFL+IDAY)                                           1342
         LDNM=LDAYNM+(IDAY-1)*8-1                                          1343
C                                                                          1344
        WRITE(SYSOUT,2) PCLSNM,(ICDAT(LPNM+I),I=1,8),                      1345
     1  TOURNM,(ICDAT(LTNM+I),I=1,8),(ICDAT(LDNM+I),I=1,8),ACT             1346
                                                                           1347
```

```
C                                                                       1348
2       FORMAT(/' *** NUMBER OF CARS IN',                               1349
     1  2(1X,8A1),' FOR',2(1X,8A1),' ON DAY ',8A1/                      1350
     1  ' *** INCREASED TO ',F5.0,' FOR PCAM CALCULATIONS')             1351
C                                                                       1352
        IF(IREAD .EQ. 1)                                                1353
     1  WRITE(SYSOUT,3) CDAT(LTOUR+ACTOFF)                              1354
3       FORMAT(' ***',F5.1,' ACTUAL CARS WERE INPUT')                   1355
C                                                                       1356
         ADDSUM=ADDSUM+ACT-CDAT(LTOUR+ACTOFF)                           1357
C                                                                       1358
        CDAT(LTOUR+ACTOFF)=ACT                                          1359
        IDERR=1                                                         1360
90      CONTINUE                                                        1361
100     CONTINUE                                                        1362
C                                                                       1363
        IF(IREAD .EQ. 1 .AND. ADDSUM .NE. 0.0)                          1364
     1  WRITE(SYSOUT,4) ADDSUM,PCLSNM,(ICDAT(LPNM+I),I=1,8),            1365
     1  (ICDAT(LDNM+I),I=1,8)                                           1366.1
4       FORMAT(' ***',F5.1,' TOTAL  CARS WERE ADDED IN',2(1X,8A1),       1367
     1  ' ON DAY ',8A1)                                                 1368.1
C                                                                       1369
        IF(IDERR .EQ. 0) RETURN                                         1370
C                                                                       1371
C            REDETERMINE EFFECTIVE CARS IF CHANGE IN ACTUAL CARS        1372
C                                                                       1373
        CALL SBLACT(LPCT,LDAY)                                          1374
        CALL SBLEF(LPCT,LDAY)                                           1375
        RETURN                                                          1376
        END                                                             1377
```


SUBROUTINE DISP

Subroutine DISP carries out the DISP command. Its function is to display selected output tables for selected shifts.

DISP calls SCAN twice to get the user's table specification. Then GTDSPC is called to scan the command qualifier and SETWFL is called to determine the subset of shifts for which output will be displayed. Subroutine MRGORD determines the output order (in DORDER) that results from the previously established output order (RORDER) and the order of qualifier phrases in the current command (ORDER). We consider that the six possible permutations of the values in DORDER define six different ways of printing tables. The permutations are mapped onto unique integers by multiplying the elements of DORDER by successive powers of two. Table 8 gives the output orderings and their corresponding integer labels (after the labels have been transformed into successive integers in the range 1-6).

The integer labels are used as entries into a branch table that controls calls to routines to implement table displays in the various output orderings. In the program documented in this report, only the Day, Tour, Precinct and Precinct, Day, Tour orderings are carried out. The branch table is entered once for each table number specified by the user.

In the program documented here, only Tables 1 through 5 are valid. The subroutine provides flexibility for the user to carry out other output orders and/or output tables.

Table 8

OUTPUT ORDERINGS

Integer Label	Output Order
1	Precinct,Tour,Day
2	Tour,Precinct,Day
3	Precinct,Day,Tour
4	Day,Precinct,Tour
5	Tour,Day,Precinct
6	Day,Tour,Precinct

```
      SUBROUTINE DISP                                                    1378
C                                                                        1379
C  IMPLEMENTS THE DISP COMMAND                                           1380
C                                                                        1381
      COMMON/PNTRS/IOVRLY,IOVTR(2),                                      1382
     1NPCTDT,NPCTRD,LPCTDT,LNMLST(4),NNAMES(4),NDAYDT,LDAYNM,            1383
     2LDYRFL,NDAYRD,LDYWFL,NTRDT,LTRTB(2),LTRST,LTREND,LTRRFL,LTRNM,     1384
     3NTRRD,LTRWFL,NBLDT,LBLKTB(2),LBLRFL,NBLRD,LBLWFL,NDIVDT,NDIVRD,    1385
     4LDIVNM,LDIVFL                                                      1386
C                                                                        1387
      COMMON/STATS/T(4,8),S(4,8),PORDER(3),RORDER(3),CIND(8)             1388
      INTEGER PORDER,RORDER                                              1389
C                                                                        1390
      COMMON/SYSTEM/SYSIN,SYSOUT,IFILE,LIT                               1391
      INTEGER SYSIN,SYSOUT                                               1392
C                                                                        1393
      COMMON/KEYWDS/NKYWD,NTYPES,TYPOFF(4),KEYWD(8,30),WDTYPE(30)        1394
      INTEGER TYPOFF,WDTYPE                                              1395
      DIMENSION PCLSNM(8),DCLSNM(8),TOURNM(8)                            1396
      EQUIVALENCE (PCLSNM,KEYWD(1,4)),(DCLSNM,KEYWD(1,3)),               1397
     1(TOURNM,KEYWD(1,2))                                                1398
C                                                                        1399
      COMMON/SCODES/SEND,CMD,NUMLST,NAMLST,FSPEC,DSPEC,DUM,ERR           1400
      INTEGER SEND,CMD,FSPEC,DSPEC,DUM,ERR                               1401
C                                                                        1402
C                                                                        1403
      COMMON/STORE/TOP,BOT,RDBOT,MAXBOT,NWORDS,CDAT(6000),C2DAT(6000)    1404
      INTEGER TOP,BOT,RDBOT                                              1405
      DIMENSION ICDAT(6000),IC2DAT(6000)                                 1406
      EQUIVALENCE(ICDAT,CDAT),(IC2DAT,C2DAT)                             1407
C                                                                        1408
C                                                                        1409
      DIMENSION ORDER(3),VAL(2),DORDER(3)                                1410
      INTEGER ORDER,TYPE,VAL,DORDER                                      1411
C                                                                        1412
      COMMON/TITLES/RTITLE(60),DTITLE(60),RUNFLG,DSNFLG                  1413
      INTEGER RUNFLG,DSNFLG                                              1414
C                                                                        1415
      DIMENSION  TABHDR(5,20)                                            1416
C                                                                        1417
C     TABLE 1. PATROL CAR ACTIVITY DURING TOUR                           1418
C                                                                        1419
      DATA TABHDR(1,1),TABHDR(1,2),TABHDR(1,3),TABHDR(1,4),              1420
     1  TABHDR(1,5),TABHDR(1,6),TABHDR(1,7),TABHDR(1,8),                 1421
     1  TABHDR(1,9),TABHDR(1,10),TABHDR(1,11),TABHDR(1,12),              1422
```



```
     1  TABHDR(1,13)/                                                   1423
     1  4HTABL,4HE 1.,4H PAT,4HROL ,4HCAR ,4HACTI,4HVITY,4H DUR,        1424
C    1  4HING ,4HTOUR,4H    ,4H    ,4H    /                            1425
C                                                                      1426
C     TABLE 2. TIME ALLOCATION: CARS STARTING THE TOUR                 1427
C                                                                      1428
      DATA TABHDR(2,1),TABHDR(2,2),TABHDR(2,3),TABHDR(2,4),            1429
     1  TABHDR(2,5),TABHDR(2,6),TABHDR(2,7),TABHDR(2,8),                1430
     1  TABHDR(2,9),TABHDR(2,10),TABHDR(2,11),TABHDR(2,12),             1431
     1  TABHDR(2,13)/                                                   1432
     1  4HTABL,4HE 2.,4H TIM,4HE AL,4HLOCA,4HTION,4H: CA,4HRS S,        1433
C    1  4HTART,4HING ,4HTHE ,4HTOUR,4H    /                           1434
C                                                                      1435
C     TABLE 3. AVERAGE DELAYS OF CALLS                                 1436
C                                                                      1437
      DATA TABHDR(3,1),TABHDR(3,2),TABHDR(3,3),TABHDR(3,4),            1438
     1  TABHDR(3,5),TABHDR(3,6),TABHDR(3,7),TABHDR(3,8),                1439
     1  TABHDR(3,9),TABHDR(3,10),TABHDR(3,11),TABHDR(3,12),             1440
     1  TABHDR(3,13)/                                                   1441
     1  4HTABL,4HE 3.,4H AVE,4HRAGE,4H DEL,4HAYS ,4HOF C,4HALLS,        1442
C    1  4H    ,4H    ,4H    ,4H    ,4H    /                           1443
C                                                                      1444
C     TABLE 4. CALLS DELAYED AND PATROL INTERVAL                       1445
C                                                                      1446
      DATA TABHDR(4,1),TABHDR(4,2),TABHDR(4,3),TABHDR(4,4),            1447
     1  TABHDR(4,5),TABHDR(4,6),TABHDR(4,7),TABHDR(4,8),                1448
     1  TABHDR(4,9),TABHDR(4,10),TABHDR(4,11),TABHDR(4,12),             1449
     1  TABHDR(4,13)/                                                   1450
     1  4HTABL,4HE 4.,4H CAL,4HLS D,4HELAY,4HED A,4HND P,4HATRO,        1451
C    1  4HL IN,4HTERV,4HAL  ,4H    ,4H    /                           1452
C                                                                      1453
C     TABLE 5. STATISTICS FROM INTERNAL PCAM CALCULATIONS              1454
C                                                                      1455
      DATA TABHDR(5,1),TABHDR(5,2),TABHDR(5,3),TABHDR(5,4),            1456
     1  TABHDR(5,5),TABHDR(5,6),TABHDR(5,7),TABHDR(5,8),                1457
     1  TABHDR(5,9),TABHDR(5,10),TABHDR(5,11),TABHDR(5,12),             1458
     1  TABHDR(5,13)/                                                   1459
     1  4HTABL,4HE 5.,4H STA,4HTIST,4HICS ,4HFROM,4H INT,4HERNA,        1460
C    1  4HL PC,4HAM C,4HALCU,4HLATI,4HONS /                           1461
      LGETT=TOP                                                        1462
      TYPE=CMD                                                         1463
C                                                                      1464
C         FINDS WHICH TABLE(S) ARE TO BE DISPLAYED                     1465
C                                                                      1466
      CALL SCAN(TYPE,VAL)                                              1467
      IF(TYPE .EQ. FSPEC) GO TO 20                                     1468
10    WRITE(SYSOUT,1)                                                  1469
1     FORMAT(/' *** INVALID TABLE SPECIFICATION - REENTER.')           1470
      TOP=LGETT                                                        1471
      RETURN                                                           1472
20    KEYVAL=VAL(1)                                                    1473
                                                                       1474
```

```
      I=KEYVAL-TYPOFF(FSPEC)                                          1475
      IF(I .NE. 3 .AND. I .NE. 5) GO TO 10                            1476
      IAVG=0                                                          1477
      IF(I .EQ. 5)IAVG=1                                              1478
      CALL SCAN(TYPE,VAL)                                             1479
      IF(TYPE .NE. NUMLST) GO TO 10                                   1480
      NPARM=VAL(1)                                                    1481
      LPARM=VAL(2)                                                    1482
C                                                                     1483
C          INTERPRET QUALIFIER                                        1484
C                                                                     1485
      CALL SCAN(TYPE,VAL)                                             1486
      CALL GTDSPC(TYPE,VAL,ORDER)                                     1487
      IF(TYPE .NE. ERR) GO TO 30                                      1488
      TOP=LGETT                                                       1489
      RETURN                                                          1490
C                                                                     1491
C          SET WORK FLAGS                                             1492
C                                                                     1493
30    CALL SETWFL(IERR)                                               1494
      IF(IERR .EQ. 0) GO TO 35                                        1495
      TOP=LGETT                                                       1496
      RETURN                                                          1497
C                                                                     1498
C          SET OUTPUT ORDER                                           1499
C                                                                     1500
35    CALL MRGORD(ORDER,RORDER,DORDER)                                1501
      IORD=0                                                          1502
      DO 40 I=1,3                                                     1503
      K=2**(I-1)                                                      1504
40    IORD=IORD+K*DORDER(I)                                           1505
      IF(IORD .GT. 14) IORD=IORD-1                                    1506
      IORD=IORD-10                                                    1507
      IF(NPCTRD .EQ. 1) IORD=1                                        1508
      DO 700 I=1,NPARM                                                1509
      ITAB=ICDAT(LPARM+(I-1)*2)                                       1510
      IF(ITAB .LT. 1 .OR. ITAB .GT. 5) GO TO 700                      1511
C                                                                     1512
       IF(RUNFLG .EQ. 1) WRITE(SYSOUT,11) RTITLE                      1513
       IF( DSNFLG .EQ. 1) WRITE(SYSOUT,12) DTITLE                     1514
       WRITE(SYSOUT,3) (TABHDR(ITAB,J),J=1,13)                        1515
C          CALL ROUTINE TO DISPLAY TABLE                              1516
C                                                                     1517
      GO TO (100,200,300,400,500,600),IORD                            1518
C                                                                     1519
100   CALL DSPPDT(ITAB,IAVG)                                          1520
      GO TO 700                                                       1521
200   CALL DSPDTP(ITAB,IAVG)                                          1522
      GO TO 700                                                       1523
300   CALL DSPPDT(ITAB,IAVG)                                          1524
      GO TO 700                                                       1525
400   CALL DSPDTP(ITAB,IAVG)                                          1526
```



```
      GO TO 700                                                       1527
500   CALL DSPDTP(ITAB,IAVG)                                          1528
      GO TO 700                                                       1529
600   CALL DSPDTP(ITAB,IAVG)                                          1530
700   WRITE(SYSOUT,2)                                                 1531
2     FORMAT(2H  /2H  )                                               1532
      TOP=LGETT                                                       1533
      RETURN                                                          1534
3     FORMAT(//1HO,13A4)                                              1535
11    FORMAT(' RUN NAME: ',60A1)                                      1536
12    FORMAT(' FILE NAME: ',60A1)                                     1537
      END                                                             1538
```

SUBROUTINE DSPDTP

Subroutine DSPDTP controls DISP command output when the output order is precinct, within tour, within day. Parameter ITAB specified the output table that is to be displayed.

DSPDTP operates in a manner similar to that of DSPPDT. The primary differences are in the meanings of the different levels of aggregation and in the way the next shift to be displayed is found. The routines NXPCT, NXDAY, and NXTOUR are set up to vary the tour most quickly, followed by the day and precinct. This corresponds exactly to the output order of tour within day within precinct, but not to precinct within tour within day. Therefore, for each day selected, DSPDTP determines the number of times that NXDAY will have to be called after a precinct is located to get to the day. DSPDTP also computes the number of times that NXTOUR must be called to get a particular tour after a precinct and day have been located.

```
      SUBROUTINE DSPDTP(ITAB,IAVG)                                      1539
C                                                                       1540
C  DISPLAYS TABLE ITAB IN ORDER OF PRECINCT WITHIN TOUR WITHIN DAY      1541
C                                                                       1542
C                                                                       1543
      COMMON/STORE/TOP,BOT,RDBOT,MAXBOT,NWORDS,CDAT(6000),C2DAT(6000)   1544
      INTEGER TOP,BOT,RDBOT                                             1545
      DIMENSION ICDAT(6000),IC2DAT(6000)                                1546
      EQUIVALENCE(ICDAT,CDAT),(IC2DAT,C2DAT)                            1547
C                                                                       1548
C                                                                       1549
      COMMON/OFFSET/NMPOFF,DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,   1550
     1NWDPCT,CPDOFF,SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,NWDDY,           1551
     2QDTOFF,QXTOFF,CRTOFF,QOTOFF,QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,   1552
     3PVTOFF,HFTOFF,MFTOFF,LFTOFF,NPRIO,NWDTR,BLDOFF,QOBOFF,QNBOFF,     1553
     4EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,NWDBL            1554
C                                                                       1555
      INTEGER DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,CPDOFF,         1556
     1SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,QDTOFF,QXTOFF,CRTOFF,QOTOFF,   1557
     2QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,PVTOFF,HFTOFF,BLDOFF,          1558
     3EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,QOBOFF,QNBOFF    1559
C                                                                       1560
      COMMON/PNTRS/IOVRLY,IOVTR(2),                                     1561
     1NPCTDT,NPCTRD,LPCTDT,LNMLST(4),NNAMES(4),NDAYDT,LDAYNM,           1562
     2LDYRFL,NDAYRD,LDYWFL,NTRDT,LTRTB(2),LTRST,LTREND,LTRRFL,LTRNM,    1563
     3NTRRD,LTRWFL,NBLDT,LBLKTB(2),LBLRFL,NBLRD,LBLWFL,NDIVDT,NDIVRD,   1564
     4LDIVNM,LDIVFL                                                     1565
C                                                                       1566
```



```
      COMMON/SYSTEM/SYSIN,SYSOUT,IFILE,LIT                              1567
      INTEGER SYSIN,SYSOUT                                              1568
C                                                                       1569
      COMMON/KEYWDS/NKYWD,NTYPES,TYPOFF(4),KEYWD(8,30),WDTYPE(30)       1570
      INTEGER TYPOFF,WDTYPE                                             1571
      DIMENSION PCLSNM(8),DCLSNM(8),TOURNM(8)                           1572
      EQUIVALENCE (PCLSNM,KEYWD(1,4)),(DCLSNM,KEYWD(1,3)),              1573
     1(TOURNM,KEYWD(1,2))                                               1574
C                                                                       1575
      INTEGER COLNAM(8),NONAME(8)                                       1576
      DATA NONAME/1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H /                      1577
C                                                                       1578
      DATA BLANK/1H /,STAR/1H*/,PLUS/1H+/                               1579
C                                                                       1580
      IF(IAVG .EQ. 0) CALL MOVE(PCLSNM,COLNAM,8)                        1581
      IF(IAVG .NE. 0) CALL MOVE(NONAME,COLNAM,8)                        1582
C                                                                       1583
      NDAY=0                                                            1584
      NXTDAY=0                                                          1585
      CALL ZERO(1)                                                      1586
C                                                                       1587
C          FIND POSITION OF NEXT DAY AMONG SELECTED DAYS                1588
C                                                                       1589
10    DO 15 I=1,NDAYRD                                                  1590
      IDAY=ICDAT(LDYWFL+I-1)                                            1591
      IF(IDAY .GT. NXTDAY) GO TO 20                                     1592
15    CONTINUE                                                          1593
      GO TO 100                                                         1594
20    NXTDAY=IDAY                                                       1595
      NDAY=NDAY+1                                                       1596
      LDNM=LDAYNM+(IDAY-1)*8-1                                          1597
      CALL ZERO(2)                                                      1598
C                                                                       1599
C          FIND TYPE OF NEXT DAY                                        1600
C                                                                       1601
      NXTYPE=0                                                          1602
      NTOUR=0                                                           1603
30    DO 40 I=1,NTRRD                                                   1604
      ITYPE=ICDAT(LTRWFL+I-1)                                           1605
      IF(ITYPE .GT. NXTYPE) GO TO 50                                    1606
40    CONTINUE                                                          1607
      IF(NTOUR .GT. 1) WRITE(SYSOUT,2) (CDAT(LDNM+I),I=1,8)             1608
C          FORMAT FOR DAY HEADER                                        1609
2     FORMAT(/' DAY: ',8A1)                                             1610
C                                                                       1611
C          ACCUMULATE MEASURES FOR DAY                                  1612
C                                                                       1613
      CALL TOTAL(ITAB,2,NTOUR,1)                                        1614
      GO TO 10                                                          1615
50    NXTYPE=ITYPE                                                      1616
      NTOUR=NTOUR+1                                                     1617
      CALL ZERO(3)                                                      1618
```

```
      LTNM=LTRNM+(ITYPE-1)*8-1                                          1619
      WRITE(SYSOUT,1) (ICDAT(LDNM+I),I=1,8),TOURNM,(ICDAT(LTNM+I),      1620
     1 I=1,8)                                                           1621
C          FORMAT FOR DAY AND TOUR HEADER                             1622
1     FORMAT(/' DAY ',8A1,';',2(1X,8A1))                              1623
      IF(IAVG .EQ. 0 .OR. NTOUR .EQ. 1) CALL TITLE(ITAB,COLNAM)       1624
C                                                                     1625
C          FIND NEXT PRECINCT                                         1626
C                                                                     1627
      LPCT=0                                                          1628
      NPCT=0                                                          1629
60    LPCT=NXPCT(LPCT)                                                1630
      IF(LPCT .NE. 0) GO TO 65                                        1631
      IADD=1                                                          1632
      IF(IOVRLY .EQ. 1 .AND. NXTYPE .EQ. NTRDT) IADD=0                1633
C                                                                     1634
C          ACCUMULATE MEASURES FOR TOUR                               1635
C                                                                     1636
      CALL TOTAL(ITAB,3,NPCT,IADD)                                    1637
      GO TO 30                                                        1638
C                                                                     1639
C          GET DAY                                                    1640
C                                                                     1641
65    LDAY=0                                                          1642
      DO 70 I=1,NDAY                                                  1643
70    LDAY=NXDAY(LPCT,LDAY)                                           1644
C                                                                     1645
C          GET TOUR                                                   1646
      LTOUR=0                                                         1647
80    LTOUR=NXTOUR(LDAY,LTOUR,ITYPE)                                  1648
      IF(LTOUR .EQ. 0) GO TO 60                                       1649
      IF(ITYPE .NE. NXTYPE) GO TO 80                                  1650
      NPCT=NPCT+1                                                     1651
      FLAG=BLANK                                                      1652
      IND=ICDAT(LTOUR+TYTOFF)                                         1653
      IF(IND .LT. 3) GO TO 90                                         1654
      FLAG=STAR                                                       1655
      IF(IND .EQ. 5) FLAG=PLUS                                        1656
90    CALL ZERO(4)                                                    1657
C                                                                     1658
C          COMPUTE AND PRINT MEASURES                                 1659
C                                                                     1660
      CALL COMPTB(ITAB,LPCT,LDAY,LTOUR,ITYPE,1)                       1661
      IF(IAVG .EQ. 0) CALL PRTBL(ITAB,4,FLAG,ICDAT(LPCT+NMPOFF))      1662
      GO TO 60                                                        1663
100   IF(NDAY .LT. 2) RETURN                                          1664
      WRITE(SYSOUT,4)                                                 1665
4     FORMAT(/' GRAND')                                               1666
      CALL TOTAL(ITAB,1,NDAY,0)                                       1667
      RETURN                                                          1668
      END                                                             1669
```


SUBROUTINE DSPPDT

Subroutine DSPPDT (*disp*lay by *p*recinct, *d*ay, *t*our) is called by subroutine DISP to print DISP command output tables by precinct, day, and tour. Its parameter ITAB specifies the table number to be printed.

Subroutine ZERO is called to initialize accumulators for averages at the overall, precinct, day, and tour levels. The integer parameter of ZERO specifies the level of the accumulators to be initialized, with one (1) corresponding to the highest level (overall) and four (4) corresponding to the lowest level (tour).

Functions NXPCT, NXDAY, and NXTOUR are used to index through the precincts, days, and tours selected by the user. A labeling line and column headings for tours are printed for each precinct and day displayed.

For each tour selected, a flag (FLAG) is set that contains the character to be printed at the left of each line of table output. Another flag, IADD (which is a parameter for subroutine COMPTB), indicates whether the tour is an overlay tour, and therefore whether its measures (level 4) are to be accumulated into the measures for a day (level 3). This may seem redundant here, but the decision to include overlay tour measures in higher levels of aggregation must be made at different levels for different output orders. Subroutine COMPTB is called to compute the measures of table ITAB for the tour. Subroutine PRTBL (*print table*) prints a line of output measures (at level 4, the tour level). Subroutine TOTAL adds measures for a specified level into the accumulators for the next highest level. TOTAL also computes and prints averages for all except level 4 measures. Statistics are not printed for a level of aggregation if there is only one entry for the next lower level.

```
      SUBROUTINE DSPPDT(ITAB,IAVG)
C                                                                     1670
C  DISPLAYS TABLE ITAB IN ORDER OF TOUR WITHIN DAY WITHIN PRECINCT    1671
C                                                                     1672
      COMMON/SYSTEM/SYSIN,SYSOUT,IFILE,LIT                            1673
      INTEGER SYSIN,SYSOUT                                            1674
C                                                                     1675
      COMMON/OFFSET/NMPOFF,DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF, 1676
     1NWDPCT,CPDOFF,SPDOFF,TPDOFF,CRDOFF,STDOFF,TRDOFF,NWDDY,         1677
                                                                      1678
```

CONTINUED

1 OF 3

```
     2QDTOFF,QXTOFF,CRTOFF,QOTOFF,QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,
     3PVTOFF,HFTOFF,MFTOFF,LFTOFF,NPRIO,NWDTR,BLDOFF,QOBOFF,QNBOFF,
     4EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,NWDBL
C
        INTEGER DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,CPDOFF,
     1SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,QDTOFF,QXTOFF,CRTOFF,QOTOFF,
     2QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,PVTOFF,HFTOFF,BLDOFF,
     3EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,QOBOFF,QNBOFF
C
        COMMON/PNTRS/IOVRLY,IOVTR(2),
     1NPCTDT,NPCTRD,LPCTDT,LNMLST(4),NNAMES(4),NDAYDT,LDAYNM,
     2LDYRFL,NDAYRD,LDYWFL,NTRDT,LTRTB(2),LTRST,LTREND,LTRRFL,LTRNM,
     3NTRRD,LTRWFL,NBLDT,LBLKTB(2),LBLRFL,NBLRD,LBLWFL,NDIVDT,NDIVRD,
     4LDIVNM,LDIVFL
C
C
      COMMON/STORE/TOP,BOT,RDBOT,MAXBOT,NWORDS,CDAT(6000),C2DAT(6000)
      INTEGER TOP,BOT,RDBOT
      DIMENSION ICDAT(6000),IC2DAT(6000)
      EQUIVALENCE(ICDAT,CDAT),(IC2DAT,C2DAT)
C
C
       COMMON/KEYWDS/NKYWD,NTYPES,TYPOFF(4),KEYWD(8,30),WDTYPE(30)
       INTEGER TYPOFF,WDTYPE
       DIMENSION PCLSNM(8),DCLSNM(8),TOURNM(8)
       EQUIVALENCE (PCLSNM,KEYWD(1,4)),(DCLSNM,KEYWD(1,3)),
     1(TOURNM,KEYWD(1,2))
C
       INTEGER COLNAM(8),NONAME(8)
       DATA NONAME/1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H /
C
       DATA BLANK/1H /,STAR/1H*/,PLUS/1H+/
C
       IF(IAVG .EQ. 0) CALL MOVE(TOURNM,COLNAM,8)
       IF(IAVG .NE. 0) CALL MOVE(NONAME,COLNAM,8)
C
       CALL ZERO(1)
C
C           FIND PRECINCT
C
       NPCT=0
       LPCT=0
10     LPCT=NXPCT(LPCT)
       IF(LPCT .EQ. 0) GO TO 100
       NPCT=NPCT+1
       NDAY=0
       CALL ZERO(2)
       LDAY=0
C
C           FIND DAY
C
20     LDAY=NXDAY(LPCT,LDAY)
```



```
       IF(LDAY .EQ. 0) GO TO 80
       NDAY=NDAY+1
       IDAY=(LDAY-LPCT-DYPOFF)/NWDDY
       IDAY=ICDAT(LDYWFL+IDAY)
       LDNM=LDAYNM+(IDAY-1)*8-1
       WRITE(SYSOUT,1) PCLSNM,(ICDAT(LPCT+NMPOFF+I-1),I=1,8),
     1 (ICDAT(LDNM+I),I=1,8)
C           FORMAT FOR PRECINCT AND DAY HEADER
1      FORMAT(/' ',8A1,': ',8A1,'; DAY: ',8A1)
       IF(IAVG .EQ. 0 .OR. NDAY .EQ. 1) CALL TITLE(ITAB,COLNAM)
       NTOUR=0
       LTOUR=0
       CALL ZERO(3)
C
C           FIND TOUR
C
30     LTOUR=NXTOUR(LDAY,LTOUR,ITYPE)
       IF(LTOUR .EQ. 0) GO TO 60
       IND=ICDAT(LTOUR+TYTOFF)
       FLAG=BLANK
       IF(IND .LT. 3) GO TO 40
       FLAG=STAR
       IF(IND .EQ. 5) FLAG=PLUS
40     NTOUR=NTOUR+1
       CALL ZERO(4)
       IADD=1
       IF(IND .EQ. 5) IADD=0
C
C           COMPUTE OUTPUT MEASURES
C
       CALL COMPTB(ITAB,LPCT,LDAY,LTOUR,ITYPE,IADD)
C
C           PRINT OUTPUT MEASURES FOR SHIFT
C
       IF (IAVG .EQ. 0)CALL PRTBL(ITAB,4,FLAG,ICDAT(LTRNM+(ITYPE-1)*8))
       GO TO 30
C
C           ACCUMULATE MEASURES FOR DAYS
C
60     CALL TOTAL(ITAB,3,NTOUR,1)
       GO TO 20
80     IF(NDAY .LT. 1) RETURN
       IF(NDAY .GT. 1)WRITE(SYSOUT,2) PCLSNM,
     1(ICDAT(LPCT+NMPOFF+I-1),I=1,8)
C           FORMAT FOR PRECINCT HEADER
2      FORMAT(/' ',8A1,': ',8A1)
C
C           ACCUMULATE MEASURES FOR PRECINCTS
C
       CALL TOTAL(ITAB,2,NDAY,1)
       GO TO 10
100    IF(NPCT .LT. 2) RETURN
```

```
      WRITE(SYSOUT,3)                                                    1783
3     FORMAT(/' GRAND')                                                  1784
      CALL TOTAL(ITAB,1,NPCT,0)                                          1785
      RETURN                                                             1786
      END                                                                1787
```


SUBROUTINE GETBOT

Subroutine GETBOT (*get bot*tom) allocates storage from the "bottom" of array CDAT.[1] The input parameter N specifies the number of words of storage that are needed. The variables TOP and BOT in COMMON/STORE/ contain the subscripts of the highest free word plus one and the lower free word in CDAT, respectively. If N words of storage are available, the output parameter L is set to the subscript of the first word allocated, BOT is updated, and the storage obtained is set to zeros. If N words of storage are not available, execution is terminated.

```
      SUBROUTINE GETBOT(N,L)                                             1788
C                                                                        1789
      COMMON/SYSTEM/SYSIN,SYSOUT,IFILE,LIT                               1790
      INTEGER SYSIN,SYSOUT                                               1791
C                                                                        1792
C                                                                        1793
      COMMON/STORE/TOP,BOT,RDBOT,MAXBOT,NWORDS,CDAT(6000),C2DAT(6000)    1794
      INTEGER TOP,BOT,RDBOT                                              1795
      DIMENSION ICDAT(6000),IC2DAT(6000)                                 1796
      EQUIVALENCE(ICDAT,CDAT),(IC2DAT,C2DAT)                             1797
C                                                                        1798
C                                                                        1799
C     ALLOCATE STORAGE                                                   1800
C                                                                        1801
      L=BOT                                                              1802
      BOT=L+N                                                            1803
C                                                                        1804
C     ERROR CONDITION     INSUFFICIENT SPACE                             1805
      IF(BOT .LT. TOP) GO TO 10                                          1806
      WRITE(SYSOUT,1)                                                    1807
1     FORMAT(/' *** INSUFFICIENT STORAGE FOR TABLES OR DATA - ',         1808
     1'EXECUTION TERMINATED')                                            1809
      STOP                                                               1810
C                                                                        1811
C     SET DATA TO ZERO                                                   1812
C                                                                        1813
10    K=BOT-1                                                            1814
      DO 20 I=L,K                                                        1815
20    ICDAT(I)=0                                                         1816
      IF(BOT .GT. MAXBOT) MAXBOT=BOT                                     1817
      RETURN                                                             1818
      END                                                                1819
                                                                         1820
```

[1]See Sec. IV for a description of the storage management system.

SUBROUTINE GETTKN

Subroutine GETTKN (*get token*) obtains the next lexical element in the user's command input. Its two parameters TYPE and VAL are set to the class and value, respectively, of the element obtained. VAL is an eight-word array; its use depends on the type of element scanned, as shown in Table 9.

At entry, if TYPE indicates that an "end of command" was the last element scanned, a new record is read from the input file and scanned from its start. A new record is also read if the program encounters an ampersand while scanning for the next element.

Function LKP1 is invoked to determine the type and value of each character scanned.

Table 9

LEXICAL TYPES RETURNED FROM GETTKN

Type Identifier (TYPE)	Description	Form of VAL
LEND	End of command	--
WORD	Character string of up to eight characters, starting with a letter	Each computer word contains one character as if it were read in A1 format. WORDs are left adjusted in VAL and padded with blanks.
NUM	Number	The first word of VAL contains the integer representation of the number, and the second word contains its floating point representation.
LP	Left parenthesis	--
RP	Right parenthesis	--

Two lines of this subroutine, 1859 and 1860, must be removed to obtain an interactive program. These print out the command that has just been read, which is unnecessary and annoying if the user has typed the command into the terminal.

```
      SUBROUTINE GETTKN(TYPE,VALUE)                                      1821
C                                                                        1822
C GETS NEXT LEXICAL ELEMENT IN USER'S COMMAND INPUT                      1823
C                                                                        1824
      COMMON/SYSTEM/SYSIN,SYSOUT,IFILE,LIT                               1825
      INTEGER SYSIN,SYSOUT                                               1826
C                                                                        1827
      COMMON/LCODES/LEND,WORD,NUM,LP,RP                                  1828
      INTEGER WORD,RP                                                    1829
C                                                                        1830
      INTEGER TYPE,VALUE,CHAR,ALPHNM,DIGIT,CHARBL,CHARLP,                1831
     1CHARRP,CHARAM,CHARST                                               1832
C                                                                        1833
      DIMENSION VALUE(8),ALPHNM(38),LETTER(26),DIGIT(12)                 1834
C                                                                        1835
      EQUIVALENCE (LETTER,ALPHNM),(DIGIT,ALPHNM(27)),                    1836
     1(IXVAL,XVAL)                                                       1837
C                                                                        1838
      DATA ALPHNM(1)/1HA/,ALPHNM(2)/1HB/,ALPHNM(3)/1HC/,                 1839
     1ALPHNM(4)/1HD/,ALPHNM(5)/1HE/,ALPHNM(6)/1HF/,ALPHNM(7)/1HG/,       1840
     2ALPHNM(8)/1HH/,ALPHNM(9)/1HI/,ALPHNM(10)/1HJ/,ALPHNM(11)/1HK/,     1841
     3ALPHNM(12)/1HL/,ALPHNM(13)/1HM/,ALPHNM(14)/1HN/,ALPHNM(15)/1HO/,   1842
     4ALPHNM(16)/1HP/,ALPHNM(17)/1HQ/,ALPHNM(18)/1HR/,ALPHNM(19)/1HS/,   1843
     5ALPHNM(20)/1HT/,ALPHNM(21)/1HU/,ALPHNM(22)/1HV/,ALPHNM(23)/1HW/,   1844
     6ALPHNM(24)/1HX/,ALPHNM(25)/1HY/,ALPHNM(26)/1HZ/                    1845
C                                                                        1846
      DATA DIGIT(1)/1HO/,DIGIT(2)/1H1/,DIGIT(3)/1H2/,DIGIT(4)/1H3/,      1847
     1DIGIT(5)/1H4/,DIGIT(6)/1H5/,DIGIT(7)/1H6/,DIGIT(8)/1H7/,           1848
     2DIGIT(9)/1H8/,DIGIT(10)/1H9/,DIGIT(11)/1H./,CHARLP/1H(/,           1849
     3CHARRP/1H)/,CHARAM/1H&/,CHARBL/1H /,DIGIT(12)/1H-/,CHARST/1H*/     1850
C                                                                        1851
      COMMON/BUFFER/CARD(81),COL                                         1852
      INTEGER CARD,COL                                                   1853
C                                                                        1854
      IF(TYPE .NE. LEND .AND. COL .LT. 81) GO TO 120                     1855
100   READ(SYSIN,2) (CARD(I),I=1,80)                                     1856
2     FORMAT(80A1)                                                       1857
C     **** NEXT TWO LINES NOT NEEDED FOR INTERACTIVE MODE ****           1858
      WRITE(SYSOUT,3) (CARD(I),I=1,80)                                   1859
3     FORMAT(1H ,80A1)                                                   1860
      COL=0                                                              1861
115   COL=COL+1                                                          1862
      IF(COL .LT. 81) GO TO 120                                          1863
```

```
      TYPE=LEND
      RETURN
C
C          FIND NEXT NON-DELIMITER
C
120   CHAR=CARD(COL)
      IF(CHAR .EQ. CHARBL) GO TO 115
      I=LKP1(CHAR,ALPHNM,38)
      IF(I .NE. 0) GO TO 200
      IF(CHAR .EQ. CHARAM) GO TO 100
      IF(CHAR .EQ. CHARRP) GO TO 150
      IF(CHAR .EQ. CHARLP) GO TO 160
      IF(CHAR .NE. CHARST) GO TO 115
C
C          FOUND '*'
C
      TYPE=WORD
      DO 125 J=2,8
125   VALUE(J)=CHARBL
      VALUE(1)=CHAR
      COL=COL+1
      RETURN
C
C          FOUND RIGHT PAREN
C
150   TYPE=RP
      COL=COL+1
      RETURN
C
C          FOUND LEFT PAREN
C
160   TYPE=LP
      COL=COL+1
      RETURN
C
C          FOUND WORD
C
200   IF(I .GT. 26) GO TO 300
      TYPE=WORD
      DO 210 J=2,8
210   VALUE(J)=CHARBL
      J=0
220   J=J+1
      IF (J .GT. 8) GO TO 230
      VALUE(J)=CHAR
230   COL=COL+1
      CHAR=CARD(COL)
      IF(LKP1(CHAR,ALPHNM,38) .NE. 0) GO TO 220
      RETURN
C
C          FOUND NUM OR '-*'
C
```



```
300   TYPE=NUM
      I=I-26
      IVAL=0
      ISIGN=1
      IF(I .NE. 12) GO TO 310
      ISIGN=-1
      IF(CARD(COL+1) .NE. CHARST) GO TO 320
      COL=COL+2
      TYPE=WORD
      DO 305 J=3,8
305   VALUE(J)=CHARBL
      VALUE(1)=CHAR
      VALUE(2)=CHARST
      RETURN
C
C          GET INTEGER VALUE
C
310   IF(I .EQ. 11) GO TO 350
      IVAL=IVAL*10+I-1
320   COL=COL+1
      CHAR=CARD(COL)
      I=LKP1(CHAR,DIGIT,11)
      IF (I .NE. 0) GO TO 310
      IVAL=IVAL*ISIGN
      VALUE(1)=IVAL
      XVAL=IVAL
      VALUE(2)=IXVAL
      RETURN
C
C          GET REAL VALUE (WITH FRACTION, IF PRESENT)
C
350   XVAL=IVAL
      POWER=1.
360   COL=COL+1
      CHAR=CARD(COL)
      I=LKP1(CHAR,DIGIT,10)
      IF(I .EQ. 0) GO TO 370
      POWER=POWER*10.
      XVAL=XVAL+(I-1)/POWER
      GO TO 360
370   XVAL=XVAL*ISIGN
      VALUE(2)=IXVAL
      VALUE(1)=IVAL*ISIGN
      RETURN
      END
```

SUBROUTINE GETTOP

This subroutine operates like GETBOT, except that allocated storage is obtained from the top of array CDAT and is not initialized when allocated.

```
      SUBROUTINE GETTOP(N,L)                                              1961
C                                                                         1962
C  ALLOCATES STORAGE AT TOP OF DATA ARRAY                                 1963
C                                                                         1964
C                                                                         1965
      COMMON/STORE/TOP,BOT,RDBOT,MAXBOT,NWORDS,CDAT(6000),C2DAT(6000)     1966
      INTEGER TOP,BOT,RDBOT                                               1967
      DIMENSION ICDAT(6000),IC2DAT(6000)                                  1968
      EQUIVALENCE(ICDAT,CDAT),(IC2DAT,C2DAT)                              1969
C                                                                         1970
C                                                                         1971
      COMMON/SYSTEM/SYSIN,SYSOUT,IFILE,LIT                                1972
      INTEGER SYSIN,SYSOUT                                                1973
C                                                                         1974
      L=TOP-N                                                             1975
      TOP=L                                                               1976
      IF(TOP .GT. BOT) RETURN                                             1977
      WRITE(SYSOUT,1)                                                     1978
1     FORMAT(/' *** INSUFFICIENT TEMPORARY STORAGE - EXECUTION',          1979
     1' TERMINATED')                                                      1980
      STOP                                                                1981
      END                                                                 1982
```


SUBROUTINE GTDSPC

Subroutine GTDSPC (*get data specification*) is called to scan command qualifiers. It obtains up to four lists of names that are the user's specifications for days, tours, divisions, and precincts. The lists are stored in array CDAT. List pointers are stored in array LNMLST and list lengths are stored in array NNAMES (see Sec. IV).

GTDSPC parameters TYPE and VAL are passed to subroutine SCAN, which is called to obtain syntactic elements from the input stream. Thus, GTDSPC's calling program can determine what syntactic element followed the qualifier in the input stream. At entry, GTDSPC assumed that TYPE and VAL have been set by a previous call to SCAN so that they describe the first element of the qualifier, or the next input element if the qualifier is null.

GTDSPC also returns a three-element array (ORDER), which specifies the order of the phrase types in the qualifier (which in turn determines the DISP command default output order). The elements of ORDER correspond to phrases in the qualifier--e.g., ORDER(1) refers to the first phrase, ORDER(2) to the second phrase, ORDER(3) to the third phrase. The value of the elements of ORDER indicate the type of phrase in each position as follows: 1 = DAY phrase, 2 = TOUR phrase, 3 = DIVISION or PRECINCT phrase (the numbers are derived from the order of the keywords in table KEYWD; DIVISION and PRECINCT are considered equivalent in this context, and the second one entered is ignored).

```
      SUBROUTINE GTDSPC(TYPE,VAL,ORDER)
C                                                                         1983
C  GETS DATA SPECIFICATION BY SCANNING QUALIFIERS                         1984
C                                                                         1985
      COMMON/SYSTEM/SYSIN,SYSOUT,IFILE,LIT                                1986
      INTEGER SYSIN,SYSOUT                                                1987
C                                                                         1988
      COMMON/KEYWDS/NKYWD,NTYPES,TYPOFF(4),KEYWD(8,30),WDTYPE(30)         1989
      INTEGER TYPOFF,WDTYPE                                               1990
      DIMENSION PCLSNM(8),DCLSNM(8),TOURNM(8)                             1991
      EQUIVALENCE (PCLSNM,KEYWD(1,4)),(DCLSNM,KEYWD(1,3)),                1992
     1(TOURNM,KEYWD(1,2))                                                 1993
C                                                                         1994
      COMMON/PNTRS/IOVRLY,IOVTR(2),                                       1995
     1NPCTDT,NPCTRD,LPCTDT,LNMLST(4),NNAMES(4),NDAYDT,LDAYNM,             1996
                                                                          1997
```

```
     2LDYRFL,NDAYRD,LDYWFL,NTRDT,LTRTB(2),LTRST,LTREND,LTRRFL,LTRNM,     1998
     3NTRRD,LTRWFL,NBLDT,LBLKTB(2),LBLRFL,NBLRD,LBLWFL,NDIVDT,NDIVRD,   1999
     4LDIVNM,LDIVFL                                                     2000
C                                                                       2001
      COMMON/SCODES/SEND,CMD,NUMLST,NAMLST,FSPEC,DSPEC,DUM,ERR          2002
      INTEGER SEND,CMD,FSPEC,DSPEC,DUM,ERR                              2003
C                                                                       2004
      INTEGER TYPE,VAL,ORDER                                            2005
      DIMENSION VAL(2),ORDER(3)                                         2006
      LGETT=TOP                                                         2007
      IORDER=0                                                          2008
      DO 5 I=1,3                                                        2009
      ORDER(I)=0                                                        2010
5     NNAMES(I)=0                                                       2011
      NNAMES(4)=0                                                       2012
      GO TO 12                                                          2013
C                                                                       2014
C          GET PHRASE TYPE                                              2015
C                                                                       2016
10    CALL SCAN(TYPE,VAL)                                               2017
12    IF(TYPE .EQ. DSPEC) GO TO 15                                      2018
      IF(TYPE .EQ. ERR .OR. TYPE .EQ. FSPEC .OR. TYPE .EQ. SEND)        2019
     1 RETURN                                                           2020
      WRITE(SYSOUT,2)                                                   2021
2     FORMAT(/' *** INVALID QUALIFIER - REENTER')                       2022
      TYPE=ERR                                                          2023
      RETURN                                                            2024
15    KEYVAL=VAL(1)                                                     2025
      IT=KEYVAL-TYPOFF(DSPEC)                                           2026
C                                                                       2027
C          GET NAME LIST                                                2028
C                                                                       2029
      CALL SCAN(TYPE,VAL)                                               2030
      IF(TYPE .EQ. NAMLST) GO TO 20                                     2031
      WRITE(SYSOUT,1)(KEYWD(I,KEYVAL),I=1,8)                            2032
1     FORMAT(/' *** INVALID ',8A1,' SPECIFICATION - REENTER.')          2033
      TYPE=ERR                                                          2034
      TOP=LGETT                                                         2035
      RETURN                                                            2036
C                                                                       2037
C DETERMINE OUTPUT ORDER SPECIFIED BY THIS COMMAND                      2038
C                                                                       2039
20    LNMLST(IT)=VAL(2)                                                 2040
      NNAMES(IT)=VAL(1)                                                 2041
      IORDER=IORDER+1                                                   2042
      IF(IORDER .GT. 3) GO TO 10                                        2043
      IF(IT .EQ. 4) IT=3                                                2044
      IF(LKP1(IT,ORDER,3) .NE. 0) GO TO 10                              2045
      ORDER(IORDER)=IT                                                  2046
      GO TO 10                                                          2047
      END                                                               2048
```


SUBROUTINE HEAD

Subroutine HEAD implements the HEADR command. Its function is to extract a runname from the command input and store it for later printing on output reports.

The header information (up to 60 characters) is first stored in an input buffer. Subroutine MOVE is then called to move the information into the vector RTITLE.

```
      SUBROUTINE HEAD                                                   2049
C                                                                       2050
C SUBROUTINE SETS UP A USER SUPPLIED HEADER FOR OUTPUT BY               2051
C EXTRACTING THE REMAINDER OF THE COMMAND LINE FOLLOWING                2052
C THE 'HEADR' COMMAND                                                   2053
C                                                                       2054
      COMMON/TITLES/RTITLE(60),DTITLE(60),RUNFLG,DSNFLG                 2055
      INTEGER RUNFLG,DSNFLG                                             2056
C                                                                       2057
      COMMON/BUFFER/CARD(81),COL                                        2058
      INTEGER CARD,COL                                                  2059
C                                                                       2060
      CALL MOVE(CARD(COL),RTITLE,60)                                    2061
      RUNFLG=1                                                          2062
      RETURN                                                            2063
      END                                                               2064
```

SUBROUTINE INIT

This subroutine performs initialization tasks for PCAM. It is called only once (from MAIN).

The initialization tasks consist primarily of reading control information from the database and allocating storage for tables whose dimensions will not change during program execution. In addition, starting hours for blocks and starting and ending hours for tours are computed.

The array KEYWD, which contains all command language keywords, is initialized by writing literals on file LIT. (This is a variable name containing a FORTRAN unit number. See Sec. I.) File LIT is read back under A format. This procedure eliminates the need for a DATA statement, which some compilers restrict to initializing only one array element per entry, and simplifies modification of keywords. In addition, the program determines the relative position among keywords (in KEYWD) of the first keyword of each "syntactic type."[2] The relative positions of the first keywords of each type are in array TYPOFF.

```
      SUBROUTINE INIT                                                    2065
C                                                                        2066
C SUBROUTINE TO INITIALIZE PERMANENT TABLES, ETC.                        2067
C                                                                        2068
      COMMON/SYSTEM/SYSIN,SYSOUT,IFILE,LIT                              2069
      INTEGER SYSIN,SYSOUT                                               2070
C                                                                        2071
      COMMON/PNTRS/IOVRLY,IOVTR(2),                                      2072
     1NPCTDT,NPCTRD,LPCTDT,LNMLST(4),NNAMES(4),NDAYDT,LDAYNM,            2073
     2LDYRFL,NDAYRD,LDYWFL,NTRDT,LTRTB(2),LTRST,LTREND,LTRRFL,LTRNM,     2074
     3NTRRD,LTRWFL,NBLDT,LBLKTB(2),LBLRFL,NBLRD,LBLWFL,NDIVDT,NDIVRD,    2075
     4LDIVNM,LDIVFL                                                      2076
C                                                                        2077
      COMMON/KEYWDS/NKYWD,NTYPES,TYPOFF(4),KEYWD(8,30),WDTYPE(30)        2078
      INTEGER TYPOFF,WDTYPE                                              2079
      DIMENSION PCLSNM(8),DCLSNM(8),TOURNM(8)                            2080
      EQUIVALENCE (PCLSNM,KEYWD(1,4)),(DCLSNM,KEYWD(1,3)),               2081
     1(TOURNM,KEYWD(1,2))                                                2082
C                                                                        2083
C                                                                        2084
      COMMON/STORE/TOP,BOT,RDBOT,MAXBOT,NWORDS,CDAT(6000),C2DAT(6000)    2085
      INTEGER TOP,BOT,RDBOT                                              2086
      DIMENSION ICDAT(6000),IC2DAT(6000)                                 2087
```

[2]See description of subroutine SCAN for syntactic types.


```
      EQUIVALENCE(ICDAT,CDAT),(IC2DAT,C2DAT)
C                                                                        2088
      COMMON/OPTION/NOERCK                                               2089
C                                                                        2090
      COMMON/TITLES/RTITLE(60),DTITLE(60),RUNFLG,DSNFLG                  2091
      INTEGER RUNFLG,DSNFLG                                              2092
C                                                                        2093
      WRITE(SYSOUT,6)                                                    2094
C                                                                        2095
C                                                                        2096
C     WRITE KEYWORD SCRATCH FILE                                         2097
C                                                                        2098
      WRITE(LIT,11)                                                      2099
      WRITE(LIT,12)                                                      2100
      WRITE(LIT,13)                                                      2101
      REWIND LIT                                                         2102
      READ(LIT,10) ((KEYWD(I,J),I=1,8),J=1,NKYWD)                        2103
C                                                                        2104
C                                                                        2105
C     READ CONTROL CARD FROM DATA BASE                                   2106
C                                                                        2107
      REWIND IFILE                                                       2108
      READ(IFILE,1) DCLSNM,PCLSNM,TOURNM,NDIVDT,NPCTDT,NDAYDT,NBLDT,     2109
     1NTRDT,IOVRLY,NOERCK,IC2DAT(1),IDSNFL                               2110
C                                                                        2111
      IF(IDSNFL.EQ.0) GO TO 60                                           2112
      READ(IFILE,7) (DTITLE(I),I=1,60)                                   2113
      DSNFLG=1                                                           2114
60    CONTINUE                                                           2115
C                                                                        2116
C                                                                        2117
C     ALLOCATE STORAGE SPACE                                             2118
C                                                                        2119
      N=8*NDAYDT                                                         2120
      CALL GETBOT(N,LDAYNM)                                              2121
      CALL GETBOT(NDAYDT,LDYRFL)                                         2122
      NL=LDAYNM+N-1                                                      2123
      READ(IFILE,2) (CDAT(I),I=LDAYNM,NL)                                2124
C                                                                        2125
      CALL GETBOT(NBLDT,LBLKTB(1))                                       2126
      CALL GETBOT(NBLDT,LBLKTB(2))                                       2127
      CALL GETBOT(NBLDT,LBLRFL)                                          2128
      K=LBLKTB(2)-1                                                      2129
      READ(IFILE,3) (ICDAT(K+I),I=1,NBLDT)                               2130
      ICDAT(LBLKTB(1))=1                                                 2131
C                                                                        2132
90    DO 100 I=2,NBLDT                                                   2133
100   ICDAT(LBLKTB(1)+I-1)=ICDAT(LBLKTB(2)+I-2)+1                        2134
C                                                                        2135
      N=8*NTRDT                                                          2136
      CALL GETBOT(N,LTRNM)                                               2137
      CALL GETBOT(NTRDT,LTRTB(1))                                        2138
      CALL GETBOT(NTRDT,LTRTB(2))                                        2139
```

```
      CALL GETBOT(NTRDT,LTRRFL)                                          2140
      L2=LTRNM-1                                                         2141
      DO 120 I=1,NTRDT                                                   2142
      L1=L2+1                                                            2143
      L2=L1+7                                                            2144
      J=LTRTB(1)+I-1                                                     2145
      K=LTRTB(2)+I-1                                                     2146
      READ(IFILE,4) (CDAT(L3),L3=L1,L2),ICDAT(J),ICDAT(K)                2147
120    CONTINUE                                                          2148
      CALL GETBOT(NTRDT,LTRST)                                           2149
      CALL GETBOT(NTRDT,LTREND)                                          2150
C                                                                        2151
C                                                                        2152
C       CALCULATE STARTING AND ENDING HOURS                              2153
C                                                                        2154
      DO 130 ITOUR=1,NTRDT                                               2155
      IBLK=ICDAT(LTRTB(1)+ITOUR-1)                                       2156
      ISTART=ICDAT(LBLKTB(1)+IBLK-1)                                     2157
      ICDAT(LTRST+ITOUR-1)=ISTART                                        2158
      IEND=ICDAT(LBLKTB(2)+IBLK-1)                                       2159
      ICDAT(LTREND+ITOUR-1)=IEND                                         2160
      IBLK=ICDAT(LTRTB(2)+ITOUR-1)                                       2161
      IF(IBLK .EQ. 0) GO TO 130                                          2162
      IEND=ICDAT(LBLKTB(2)+IBLK-1)                                       2163
      ICDAT(LTREND+ITOUR-1)=IEND                                         2164
130   CONTINUE                                                           2165
C                                                                        2166
      CALL GETBOT(8*NDIVDT,LDIVNM)                                       2167
      RDBOT=BOT                                                          2168
C                                                                        2169
C          ASSIGN TYPES TO KEYWORDS                                      2170
C                                                                        2171
      I=0                                                                2172
      DO 150 ITYPE=1,NTYPES                                              2173
      N=TYPOFF(ITYPE)                                                    2174
      TYPOFF(ITYPE)=I                                                    2175
      DO 140 J=1,N                                                       2176
      I=I+1                                                              2177
      IF(I .LE. NKYWD) GO TO 140                                         2178
      WRITE(SYSOUT,5) ITYPE                                              2179
      STOP                                                               2180
140   WDTYPE(I)=ITYPE                                                    2181
150   CONTINUE                                                           2182
      REWIND IFILE                                                       2183
      RETURN                                                             2184
C                                                                        2185
1     FORMAT(8A1,2X,8A1,2X,8A1,1X,I2,1X,I3,1X,I3,1X,I2,1X,I2,1X,I1,      2186
     13(1X,I1))                                                          2187
2     FORMAT(80A1)                                                       2188
3     FORMAT(24(I2,1X))                                                  2189
4     FORMAT(8A1,1X,I2,1X,I2)                                            2190
6     FORMAT(///26X,                                                     2191
```



```
     C  'PATROL CAR ALLOCATION MODEL'///)
7     FORMAT(60A1)                                                       2192
11    FORMAT(                                                            2193
     C'DAY      ',                                                      2194
     C'TOUR     ',                                                      2195
     C'DIVISION',                                                       2196
     C'PRECINCT',                                                       2197
     C'P        ',                                                      2198
     C'C        ',                                                      2199
     C'T        ',                                                      2200
     C'F        ',                                                      2201
     C'A        ',                                                      2202
     C'ADD      ')                                                      2203
12    FORMAT(                                                            2204
     C'ALOC     ',                                                      2205
     C'DISP     ',                                                      2206
     C'END      ',                                                      2207
     C'HEADR    ',                                                      2208
     C'LIST     ',                                                      2209
     C'MEET     ',                                                      2210
     C'READ     ',                                                      2211
     C'SET      ',                                                      2212
     C'WRITE    ',                                                      2213
     C'FOR      ')                                                      2214
13    FORMAT(                                                            2215
     C'CAR      ',                                                      2216
     C'CARS     ',                                                      2217
     C'TO       ',                                                      2218
     C'BY       ',                                                      2219
     C'DATA     ',                                                      2220
     C'HOUR     ',                                                      2221
     C'HOURS    ',                                                      2222
     C'ON       ')                                                      2223
10    FORMAT(80A1)                                                       2224
5     FORMAT(/' ***INTERNAL ERROR: TOO MANY KEYWORDS AT TYPE ',          2225
     C I2,' - EXECUTION TERMINATED')                                    2226
      END                                                               2227
                                                                        2228
```

FUNCTION KNSTR

Function KNSTR determines whether a given number of effective or actual cars on duty in a block of a day results in a specified constraint on an output measure being met. A function value of one (1) is returned if the constraint is met, otherwise a value of zero (0) is returned. Parameter ICNSTR specifies the output measure whose value, with EF effective cars or ACT actual cars, is to be tested against constraint value CVAL. The valid values of ICNSTR are the output measure specifications given in the MEET command description in Sec. III of the User's Manual. LPCT, LDAY, LTOUR, and LBLK are pointers to the data for the precinct, day, tour, and block for which the output measure is to be tested. IBLD is the position of the block relative to all blocks in the database (e.g., the *third* block of a day).

The output measure specified by ICNSTR is evaluated for the block, given EF effective cars. Some output measures are computed directly from available data; others are computed by function references. The resulting measure is tested against CVAL and the value of KNSTR set according to the outcome.

```
      FUNCTION KNSTR(ICNSTR,CVAL,ACT,EF,LPCT,LDAY,LTOUR,LBLK,IBLD)      2229
C                                                                        2230
C  DETERMINES WHETHER CONSTRAINT CVAL ON PERFORMANCE MEASURE             2231
C  ICNSTR IS MET BY EF EFFECTIVE CARS                                    2232
C                                                                        2233
C                                                                        2234
      COMMON/STORE/TOP,BOT,RDBOT,MAXBOT,NWORDS,CDAT(6000),C2DAT(6000)    2235
      INTEGER TOP,BOT,RDBOT                                              2236
      DIMENSION ICDAT(6000),IC2DAT(6000)                                 2237
      EQUIVALENCE(ICDAT,CDAT),(IC2DAT,C2DAT)                             2238
C                                                                        2239
      DIMENSION C1(3),C2(3),C3(3)                                        2240
C                                                                        2241
      COMMON/PNTRS/IOVRLY,IOVTR(2),                                      2242
     1NPCTDT,NPCTRD,LPCTDT,LNMLST(4),NNAMES(4),NDAYDT,LDAYNM,            2243
     2LDYRFL,NDAYRD,LDYWFL,NTRDT,LTRTB(2),LTRST,LTREND,LTRRFL,LTRNM,     2244
     3NTRRD,LTRWFL,NBLDT,LBLKTB(2),LBLRFL,NBLRD,LBLWFL,NDIVDT,NDIVRD,    2245
     4LDIVNM,LDIVFL                                                      2246
C                                                                        2247
      COMMON/OFFSET/NMPOFF,DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,    2248
     1NWDPCT,CPDOFF,SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,NWDDY,            2249
     2QDTOFF,QXTOFF,CRTOFF,QOTOFF,QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,    2250
     3PVTOFF,HFTOFF,MFTOFF,LFTOFF,NPRIO,NWDTR,BLDOFF,QOBOFF,QNBOFF,      2251
     4EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,NWDBL             2252
```



```
C                                                                        2253
      INTEGER DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,CPDOFF,          2254
     1SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,QDTOFF,QXTOFF,CRTOFF,QOTOFF,    2255
     2QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,PVTOFF,HFTOFF,BLDOFF,           2256
     3EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,QOBOFF,QNBOFF     2257
C                                                                        2258
      KNSTR=0                                                            2259
      ISTART=ICDAT(LBLKTB(1)+IBLD-1)                                     2260
      IEND=ICDAT(LBLKTB(2)+IBLD-1)                                       2261
      LCR=LDAY+CRDOFF                                                    2262
      LST=LDAY+STDOFF                                                    2263
      LFR=LTOUR+HFTOFF                                                   2264
C                                                                        2265
      DO 50  I=1,3                                                       2266
      C1(I)=C2DAT(LBLK+QDTOFF+I-1)                                       2267
      C2(I)=C2DAT(LBLK+QOTOFF+I-1)                                       2268
   50 C3(I)=C2DAT(LBLK+TYTOFF+I-1)                                       2269
C                                                                        2270
      BLKLN=IEND-ISTART+1                                                2271
      RV=CDAT(LTOUR+RVTOFF)                                              2272
C                                                                        2273
      GO TO (100,200,300,400,500,600,700,800,900,1000,1100,1200,         2274
     1 1300),ICNSTR                                                      2275
C                                                                        2276
  100 X=CDAT(LBLK+AWBOFF)/ACT*100.0                                      2277
      IF(X .LE. CVAL) KNSTR=1                                            2278
      RETURN                                                             2279
C                                                                        2280
  200 RV=CDAT(LTOUR+RVTOFF)                                              2281
      X=AVTT(ISTART,IEND,LPCT,LDAY,RV,EF)                                2282
      IF( X .LE. CVAL) KNSTR=1                                           2283
      RETURN                                                             2284
  300 X=EF-CDAT(LBLK+AWBOFF)                                             2285
      IF(X .GE. CVAL) KNSTR=1                                            2286
      RETURN                                                             2287
C                                                                        2288
  400 RETURN                                                             2289
C                                                                        2290
  500 X=(EF-CDAT(LBLK+AWBOFF))*CDAT(LTOUR+PVTOFF)/CDAT(LPCT+SMPOFF)      2291
C                                                                        2292
      X=1.0/X*60.0                                                       2293
      IF( X .LE. CVAL) KNSTR=1                                           2294
      RETURN                                                             2295
  600 RETURN                                                             2296
  700 X=OBJF1(0,ISTART,IEND,LPCT,LCR,LST,LFR,RV,EF,C1,C2,C3)/            2297
     1 CDAT(LBLK+CRBOFF)                                                 2298
C                                                                        2299
      X=X*100.0                                                          2300
C                                                                        2301
      IF(X .LE. CVAL) KNSTR=1                                            2302
      RETURN                                                             2303
                                                                         2304
```

```
C
C      800 AVERAGE DELAY FOR PRIORITY 2 CALLS (MINUTES)
C      900 AVERAGE DELAY FOR PRIORITY 3 CALLS (MINUTES)
C     1000 AVERAGE DELAY FOR ALL CALLS (MINUTES)
C     1100 TOTAL WAIT (QUEUING+TRAVEL) PRIORITY 2 CALLS (MINUTES)
C     1200 TOTAL WAIT (QUEUING+TRAVEL) PRIORITY 3 CALLS (MINUTES)
C     1300 TOTAL WAIT (QUEUING+TRAVEL) ALL CALLS (MINUTES
C
800     N=2
        GO TO 910
900     N=3
910     X=OBJF2(N,ISTART,IEND,LPCT,LCR,LST,LFR,RV,EF,C1,C2,C3)/
     1  (CDAT(LBLK+CRBOFF)*CDAT(LFR+N-1))
        IF(X .LE. CVAL) KNSTR=1
        RETURN
1000    X=OBJF2(0,ISTART,IEND,LPCT,LCR,LST,LFR,RV,EF,C1,C2,C3)/
     1  CDAT(LBLK+CRBOFF)
        IF(X .LE. CVAL) KNSTR=1
        RETURN
1100    N=2
        GO TO 1210
1200    N=3
1210    X=OBJF3(N,ISTART,IEND,LPCT,LCR,LST,LFR,RV,EF,C1,C2,C3)/
     1  (CDAT(LBLK+CRBOFF)*CDAT(LFR+N-1))
        IF(X .LE. CVAL) KNSTR=1
        RETURN
1300    X=OBJF3(0,ISTART,IEND,LPCT,LCR,LST,LFR,RV,EF,C1,C2,C3)/
     1  CDAT(LBLK+CRBOFF)
        IF(X .LE. CVAL) KNSTR=1
        RETURN
C
        END
```


SUBROUTINE LIST

Subroutine LIST carries out the LIST command. It prints input data (and some derived values) for selected precincts, days, and tours.

Subroutine GTDSPC is called to scan the qualifier and SETWFL is called to define the subset of precincts, days, and tours for which data will be listed. Function NXPCT is called to set a pointer (LPCT) to the data for the next precinct selected. A pointer value of zero at entry to NXPCT requests the first precinct selected; a pointer value of zero returned from NXPCT means no more precincts have been selected. The name, area, street miles, and unavailability parameters are printed for each precinct selected.

After a precinct pointer has been obtained and the data for the precinct printed, function NXDAY is called to find the days for which data are to be listed. As with NXPCT, the value of the day data pointer (LDAY) at entry to NXDAY indicates whether the first day for a precinct is to be located, and the value of the day pointer returned from NXDAY is zero if there are no more days selected. For each selected day its name, call rate parameter, and service time parameter are printed; in addition, column headings are printed for the tour data that follow.

Function NXTOUR is used in the same manner as NXPCT and NXDAY to index through the tours of each day. For each tour selected, LIST computes average call rate and service time over all its hours and the average number of effective cars in its blocks. These are printed along with the tour name, actual cars assigned, response speed, patrol speed, and the fraction of calls in each priority class.

```
$$$SUBROUTINE LIST
SUBROUTINE LIST
C  IMPLEMENTS THE LIST COMMAND
C
C
      COMMON/STORE/TOP,BOT,RDBOT,MAXBOT,NWORDS,CDAT(6000),C2DAT(6000)
      INTEGER TOP,BOT,RDBOT
      DIMENSION ICDAT(6000),IC2DAT(6000)
      EQUIVALENCE(ICDAT,CDAT),(IC2DAT,C2DAT)
C
C
```

```
      2347 COMMON/SYSTEM/SYSIN,SYSOUT,IFILE,LIT
      2348 INTEGER SYSIN,SYSOUT
2349 C
      2350 COMMON/KEYWDS/NKYWD,NTYPES,TYPOFF(4),KEYWD(8,30),WDTYPE(30)
      2351 INTEGER TYPOFF,WDTYPE
      2352 DIMENSION PCLSNM(8),DCLSNM(8),TOURNM(8)
      2353 EQUIVALENCE (PCLSNM,KEYWD(1,4)),(DCLSNM,KEYWD(1,3)),
     2354 1(TOURNM,KEYWD(1,2))
2355 C
      2356 COMMON/OFFSET/NMPOFF,DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,
     2357 1NWDPCT,CPDOFF,SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,NWDDY,
     2358 2QDTOFF,QXTOFF,CRTOFF,QOTOFF,QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,
     2359 3PVTOFF,HFTOFF,MFTOFF,LFTOFF,NPRIO,NWDTR,BLDOFF,QOBOFF,QNBOFF,
     2360 4EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,NWDBL
2361 C
      2362 INTEGER DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,CPDOFF,
     2363 1SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,QDTOFF,QXTOFF,CRTOFF,QOTOFF,
     2364 2QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,PVTOFF,HFTOFF,BLDOFF,
     2365 3EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,QOBOFF,QNBOFF
2366 C
      2367 COMMON/PNTRS/IOVRLY,IOVTR(2),
     2368 1NPCTDT,NPCTRD,LPCTDT,LNMLST(4),NNAMES(4),NDAYDT,LDAYNM,
     2369 2LDYRFL,NDAYRD,LDYWFL,NTRDT,LTRTB(2),LTRST,LTREND,LTRRFL,LTRNM,
     2370 3NTRRD,LTRWFL,NBLDT,LBLKTB(2),LBLRFL,NBLRD,LBLWFL,NDIVDT,NDIVRD,
     2371 4LDIVNM,LDIVFL
2372 C
      2373 COMMON/SCODES/SEND,CMD,NUMLST,NAMLST,FSPEC,DSPEC,DUM,ERR
      2374 INTEGER SEND,CMD,FSPEC,DSPEC,DUM,ERR
2375 C
       2376 COMMON/TITLES/RTITLE(60),DTITLE(60),RUNFLG,DSNFLG
       2377 INTEGER RUNFLG,DSNFLG
2378 C
      2379 DIMENSION VAL(2),ORDER(3)
      2380 INTEGER TYPE,VAL
2381 C
       2382 IF(RUNFLG .EQ. 1) WRITE(SYSOUT,11) RTITLE
       2383 IF( DSNFLG .EQ. 1) WRITE(SYSOUT,12) DTITLE
2384 C
      2385 LGETT=TOP
      2386 TYPE=CMD
2387 C
2388 C           INTERPRETS QUALIFIER OF LIST COMMAND
2389 C
      2390 CALL SCAN(TYPE,VAL)
      2391 CALL GTDSPC(TYPE,VAL,ORDER)
      2392 IF(TYPE .NE. ERR) GO TO 10
      2393 TOP=LGETT
      2394 RETURN
2395 10    CALL SETWFL(IERR)
      2396 IF(IERR .EQ. 0) GO TO 15
      2397 TOP=LGETT
      2398 RETURN
2399 C
```



```
2400 C           FIND NEXT PRECINCT, WRITE HEADER INFORMATION
2401 C
2402 15    LPCT=0
2403 20    LPCT=NXPCT(LPCT)
      2404 IF(LPCT .NE. 0) GO TO 30
      2405 TOP=LGETT
      2406 RETURN
2407 C
2408 30    CONTINUE
2409 C
     2410 WRITE(SYSOUT,1) PCLSNM, (ICDAT(LPCT+NMPOFF+I-1),I=1,8),
    2411 1 CDAT(LPCT+ARPOFF),CDAT(LPCT+SMPOFF),CDAT(LPCT+B2POFF),
    2412 2 CDAT(LPCT+B1POFF)
2413 C
2414 C           FIND NEXT DAY.  LIST HEADER INFORMATION.
2415 C
2416 LDAY=0
      LDAY=NXDAY(LPCT,LDAY)                                    2417 40
      IF(LDAY .EQ. 0) GO TO 20                                      2418
      IDAY=(LDAY-LPCT-DYPOFF)/NWDDY                                 2419
      IDAY=ICDAT(LDYWFL+IDAY)                                       2420
LDNM=LDAYNM+(IDAY-1)*8-1                                            2421
     2423 WRITE(SYSOUT,2) (ICDAT(LDNM+I),I=1,8),CDAT(LDAY+CPDOFF),  2422 C
    2424 1 CDAT(LDAY+SPDOFF),TOURNM
2425 C
2426 C           FIND NEXT TOUR.  CALCULATE AVERAGE CALL RATE,
2427 C           SERVICE TIME, EFFECTIVE CARS
2428 C
      2429 LTOUR=0
2430 50    LTOUR=NXTOUR(LDAY,LTOUR,ITYPE)
      2431 IF(LTOUR .EQ. 0) GO TO 40
      2432 LTNM=LTRNM+(ITYPE-1)*8-1
      2433 IF(ICDAT(LTOUR+TYTOFF) .NE. 5) GO TO 55
      2434 WRITE(SYSOUT,3) (ICDAT(LTNM+I),I=1,8),CDAT(LTOUR+ACTOFF)
      2435 GO TO 40
2436 55    ISTART=ICDAT(LTRST+ITYPE-1)
      2437 IEND=ICDAT(LTREND+ITYPE-1)
      2438 ST=0.
      2439 DO 60 I=ISTART,IEND
2440 60    ST=ST+CDAT(LDAY+STDOFF+I-1)
      2441 TOURLN=IEND-ISTART+1
      2442 ST=ST*60./TOURLN
      2443 CR=CDAT(LTOUR+CRTOFF)/TOURLN
      2444 EF=0.
      2445 DO 70 IBLK=1,2
      2446 IBDT=ICDAT(LTRTB(IBLK)+ITYPE-1)
      2447 IF(IBDT .LT. 1) GO TO 70
      2448 IBRD=ICDAT(LBLRFL+IBDT-1)
      2449 LBLK=LDAY+BLDOFF+(IBRD-1)*NWDBL
      2450 BLKLN=ICDAT(LBLKTB(2)+IBDT-1)-ICDAT(LBLKTB(1)+IBDT-1)+1
      2451 EF=EF+BLKLN*CDAT(LBLK+EFBOFF)
2452 70    CONTINUE
      2453 EF=EF/TOURLN
```

```
      2454 WRITE(SYSOUT,3) (ICDAT(LTNM+I),I=1,8),CDAT(LTOUR+ACTOFF),
     2455 1 EF,CDAT(LTOUR+RVTOFF),
     2456 1CDAT(LTOUR+PVTOFF), ST,CR,(CDAT(LTOUR+HFTOFF+I-1),I=1,3)
      2457 GO TO 50
2458 C            FORMAT OF PRECINCT HEADER
2459 1       FORMAT(/,1H ,8A1,2H: ,8A1,'; AREA=',F5.1,'; STREET MILES=',
     2460 1 F5.1,'; B2=',F5.3,'; B1=',F5.3)
2461 C            FORMAT FOR DAY HEADER AND COLUMN LABELS
2462 2       FORMAT(/'   DAY: ',8A1,'; CALL RATE PARM=',F5.2,'; SERVICE
     TIME' 2463 1 ,' PARM=',F5.2//21X,
     2464 2' AVG.              AVG.  AVG.  FRAC.  FRAC.  FRAC.'/16X,
     2465 3'ACT.  EFF.    SPEED     SERV  CALL  OF P1  OF P2  OF P3'/
     2466 3 6X,8A1, 2X,
     2467 4'CARS  CARS  RSP.  PTL.  TIME  RATE  CALLS  CALLS  CALLS')
2468 C            FORMAT FOR ENTRIES IN COLUMNS
2469 3       FORMAT(6X,8A1,6(2X,F4.1),3(2X,F5.3))
2470 11       FORMAT('0RUN NAME: ',60A1)
2471 12       FORMAT(' FILE NAME: ',60A1)
      2472 END
2473
```


FUNCTION LKP1

Function LKP1 determines the position of a one-word argument in a list. Parameter LIST is the list to be searched for IARG. N is the number of entries in LIST. The function value returned is the position of IARG in LIST or zero if IARG is not found in LIST.

```
      FUNCTION LKP1(IARG,LIST,N)                                        2474
C                                                                       2475
C  DETERMINES WHETHER IARG IS IN LIST, AND, IF SO, WHERE                2476
C                                                                       2477
      DIMENSION LIST(N)                                                 2478
      LKP1=0                                                            2479
      IF(N .EQ. 0) RETURN                                               2480
      DO 10 I=1,N                                                       2481
      IF(IARG .EQ. LIST(I)) GO TO 20                                    2482
10    CONTINUE                                                          2483
      RETURN                                                            2484
20    LKP1=I                                                            2485
      RETURN                                                            2486
      END                                                               2487
```

FUNCTION LKP8

Function LKP8 is called to determine the position of an eight-character name in a list of names. ARG is the name to be found, LIST is the list to be searched, and N is the number of entries in LIST.

The value returned is the position of ARG in LIST, or zero if ARG is not in LIST. Note that this function is frequently invoked with both ARG and LIST as parts of one-dimensional arrays.

```
      FUNCTION LKP8(ARG,LIST,N)                                    2488
C                                                                  2489
C  DETERMINES WHETHER ARG IS IN LIST, AND, IF SO, WHERE            2490
C                                                                  2491
      REAL LIST                                                    2492
      DIMENSION ARG(8),LIST(8,N)                                   2493
      L=0                                                          2494
      LKP8=0                                                       2495
      NT=N                                                         2496
10    IF(NT .EQ. 0) RETURN                                         2497
      L=L+1                                                        2498
      DO 20 I=1,8                                                  2499
      IF(ARG(I) .NE. LIST(I,L)) GO TO 30                           2500
20    CONTINUE                                                     2501
      LKP8=L                                                       2502
      RETURN                                                       2503
30    NT=NT-1                                                      2504
      GO TO 10                                                     2505
      END                                                          2506
```


SUBROUTINE MEET

Subroutine MEET carries out the MEET command. Its function is to assign enough cars to all shifts within its scope that a user-specified set of constraints on selected performance measures is met.

At entry, successive calls to subroutine SCAN get pointers to the list of output measures (LPARM) and the list of constraint values (LVAL). If all of the output measure specifications are valid and the number of constraint values matches the number of output measure specifications, GTDSPC is called to scan the MEET command qualifier. Subroutine SETWFL sets the "work" flags for days and tours and subroutine CKOVR (*check overlay*) insures that, if an overlay tour has been specified in the qualifier, then the overlaid tours have also been specified.

MEET then indexes through all selected precincts, days, and tours. The blocks of each shift thus selected are dealt with independently. If a block has not been within the scope of a previous MEET, ADD, or ALOC command since the last READ command (ICDAT(LBLK+CTBOFF) less than 0), enough cars are assigned to the block to handle its cfs workload.

Starting with either the current assignment or the minimum assignment, the number of cars in a block is increased as necessary to meet each specified constraint in turn. Function KNSTR determines whether or not a particular constraint has been met by a given number of effective cars (KNSTR is not used for minimum manning level--constraint 6).

When constraints have been met for all blocks of all tours of a day, subroutine STRCAR (*set tour cars*) is called to obtain a feasible allocation of cars to the tours of the day that will result in the required number of cars in each block (see Sec. III). Then SBLACT (*set block actual cars*) is called to convert this tour allocation to a block allocation (see Sec. III) and SBLEF (*set block effective cars*) is called to determine the resulting number of effective cars in each block.

MEET returns when all constraints have been met for all blocks of all shifts within the scope of the command.

```
      SUBROUTINE MEET                                                   2572
C                                                                       2573
C  DETERMINES CAR REQUIREMENTS TO MEET SPECIFIED                        2574
C CONSTRAINTS.                                                          2575
C                                                                       2576
C                                                                       2577
      COMMON/STORE/TOP,BOT,RDBOT,MAXBOT,NWORDS,CDAT(6000),C2DAT(6000)   2578
      INTEGER TOP,BOT,RDBOT                                             2579
      DIMENSION ICDAT(6000),IC2DAT(6000)                                2580
      EQUIVALENCE(ICDAT,CDAT),(IC2DAT,C2DAT)                            2581
C                                                                       2582
C                                                                       2583
      COMMON/PNTRS/IOVRLY,IOVTR(2),                                     2584
     1NPCTDT,NPCTRD,LPCTDT,LNMLST(4),NNAMES(4),NDAYDT,LDAYNM,           2585
     2LDYRFL,NDAYRD,LDYWFL,NTRDT,LTRTB(2),LTRST,LTREND,LTRRFL,LTRNM,    2586
     3NTRRD,LTRWFL,NBLDT,LBLKTB(2),LBLRFL,NBLRD,LBLWFL,NDIVDT,NDIVRD,   2587
     4LDIVNM,LDIVFL                                                     2588
C                                                                       2589
      COMMON/OFFSET/NMPOFF,DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,   2590
     1NWDPCT,CPDOFF,SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,NWDDY,           2591
     2QDTOFF,QXTOFF,CRTOFF,QOTOFF,QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,   2592
     3PVTOFF,HFTOFF,MFTOFF,LFTOFF,NPRIO,NWDTR,BLDOFF,QOBOFF,QNBOFF,     2593
     4EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,NWDBL            2594
C                                                                       2595
      INTEGER DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,CPDOFF,         2596
     1SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,QDTOFF,QXTOFF,CRTOFF,QOTOFF,   2597
     2QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,PVTOFF,HFTOFF,BLDOFF,          2598
     3EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,QOBOFF,QNBOFF    2599
C                                                                       2600
      COMMON/SYSTEM/SYSIN,SYSOUT,IFILE,LIT                              2601
      INTEGER SYSIN,SYSOUT                                              2602
C                                                                       2603
      COMMON/KEYWDS/NKYWD,NTYPES,TYPOFF(4),KEYWD(8,30),WDTYPE(30)       2604
      INTEGER TYPOFF,WDTYPE                                             2605
      DIMENSION PCLSNM(8),DCLSNM(8),TOURNM(8)                           2606
      EQUIVALENCE (PCLSNM,KEYWD(1,4)),(DCLSNM,KEYWD(1,3)),              2607
     1(TOURNM,KEYWD(1,2))                                               2608
C                                                                       2609
      COMMON/SCODES/SEND,CMD,NUMLST,NAMLST,FSPEC,DSPEC,DUM,ERR          2610
      INTEGER SEND,CMD,FSPEC,DSPEC,DUM,ERR                              2611
C                                                                       2612
      INTEGER TYPE,VAL                                                  2613
      DIMENSION VAL(2),ORDER(3)                                         2614
       ISMFLG = IC2DAT(1)                                               2615
      LGETT=TOP                                                         2616
      TYPE=CMD                                                          2617
C                                                                       2618
C           GET CONSTRAINT SPECIFICATIONS, CHECK VALIDITY               2619
C                                                                       2620
      CALL SCAN(TYPE,VAL)                                               2621
      IF(TYPE .EQ. FSPEC) GO TO 20                                      2622
10    WRITE(SYSOUT,1)                                                   2623
```



```
1     FORMAT(/' *** INVALID CONSTRAINT SPECIFICATION - REENTER')        2624
      TOP=LGETT                                                         2625
      RETURN                                                            2626
20    KEYVAL=VAL(1)                                                     2627
      I=KEYVAL-TYPOFF(FSPEC)                                            2628
      IF(I .NE. 2) GO TO 10                                             2629
      CALL SCAN(TYPE,VAL)                                               2630
      IF(TYPE .NE. NUMLST) GO TO 10                                     2631
      NPARM=VAL(1)                                                      2632
      LPARM=VAL(2)                                                      2633
      DO 25 IPARM=1,NPARM                                               2634
      I=ICDAT(LPARM+(IPARM-1)*2)                                        2635
C                                                                       2636
      IF(I .GT. 0 .AND. I .LT. 14) GO TO 25                             2637
C                                                                       2638
      WRITE(SYSOUT,2) I                                                 2639
2     FORMAT(/' *** INVALID CONSTRAINT NUMBER : ',I4,' - REENTER')      2640
      TOP=LGETT                                                         2641
      RETURN                                                            2642
25    CONTINUE                                                          2643
C                                                                       2644
C           GET CONSTRAINT VALUES                                       2645
C                                                                       2646
      CALL SCAN(TYPE,VAL)                                               2647
      IF(TYPE .EQ. NUMLST) GO TO 30                                     2648
      WRITE(SYSOUT,3)                                                   2649
3     FORMAT(/' *** INVALID CONSTRAINT VALUE(S) - REENTER')             2650
      TOP=LGETT                                                         2651
      RETURN                                                            2652
30    NVAL=VAL(1)                                                       2653
      LVAL=VAL(2)                                                       2654
      IF(NVAL .EQ. NPARM) GO TO 40                                      2655
      WRITE(SYSOUT,4)                                                   2656
4     FORMAT(/' *** NUMBER OF VALUES DOES NOT MATCH NUMBER OF CONS',    2657
     1 'TRAINTS - REENTER')                                             2658
      TOP=LGETT                                                         2659
      RETURN                                                            2660
C                                                                       2661
C           SCAN QUALIFIER                                              2662
C                                                                       2663
40    CALL SCAN(TYPE,VAL)                                               2664
      CALL GTDSPC(TYPE,VAL,ORDER)                                       2665
      IF(TYPE .NE. ERR) GO TO 50                                        2666
      TOP=LGETT                                                         2667
      RETURN                                                            2668
C                                                                       2669
C           SET WORK FLAGS                                              2670
C                                                                       2671
50    CALL SETWFL(IERR)                                                 2672
      IF(IERR .EQ. 0) GO TO 55                                          2673
      TOP=LGETT                                                         2674
      RETURN                                                            2675
```

```
C
C            INSURE THAT OVERLAY SEGMENT IS COMPLETE OR NOT INCLUDED
C
55      CALL CKOVR(IERR)
        IF(IERR .EQ. 0) GO TO 60
        TOP=LGETT
        RETURN
60      LPCT=0
        NTOT=0
100     LPCT=NXPCT(LPCT)
        IF(LPCT .NE. 0) GO TO 110
        WRITE(SYSOUT,6) NTOT
6       FORMAT(/'   ',I4,' CAR HOURS ALLOCATED.')
        TOP=LGETT
        RETURN
110     B1=CDAT(LPCT+B1POFF)
        B2=CDAT(LPCT+B2POFF)
        LDAY=0
120     LDAY =NXDAY(LPCT,LDAY)
        IF(LDAY .EQ. 0) GO TO 100
130     LTOUR=0
140     LTOUR=NXTOUR(LDAY,LTOUR,ITYPE)
        IF(LTOUR .NE. 0 .AND. ICDAT(LTOUR+TYTOFF) .NE. 5) GO TO 160
C
C            ASSIGN CARS TO THE TOURS OF A DAY SO THAT BLOCK
C            REQUIREMENTS ARE MET
C
        CALL STRCAR(LDAY,CARHRS)
        NTOT=NTOT+CARHRS
C
C            DETERMINE BLOCK ASSIGNMENTS FROM TOUR ASSIGNMENTS
C
        CALL SBLACT(LPCT,LDAY)
        CALL SBLEF(LPCT,LDAY)
        GO TO 120
160     DO 220 IBLK=1,2
        IBLD=ICDAT(LTRTB(IBLK)+ITYPE-1)
        IF(IBLD .EQ. 0) GO TO 220
        IBLR=ICDAT(LBLRFL+IBLD-1)
        LBLK=LDAY+BLDOFF+(IBLR-1)*NWDBL
        AWL=CDAT(LBLK+AWBOFF)
C
        IF(ICDAT(LBLK+CTBOFF) .LT. 0) GO TO 165
        EF=CDAT(LBLK+EFBOFF)
        ACT=CDAT(LBLK+ACBOFF)
        GO TO 170
C
C            DETERMINE MINIMUM BLOCK REQUIREMENTS
C
165     EF= INT(CDAT(LBLK+RMBOFF)+1.0001)
        IF(ISMFLG .EQ. 1) EF=INT(CDAT(LBLK+AWBOFF)+1.0001)
        ACT=CEIL((EF+B1*AWL)/(1.-B2))
```



```
        CDAT(LBLK+ACBOFF)=ACT
        EF=ACT*(1.-((B1*AWL/ACT)+B2))
        CDAT(LBLK+EFBOFF)=EF
        ICDAT(LBLK+CTBOFF)=0
C
C            INSURE THAT ALL CONSTRAINTS ARE MET FOR EACH BLOCK
C
170     DO 200 IPARM=1,NPARM
        IP=ICDAT(LPARM+(IPARM-1)*2)
C
        IF(IP .EQ. 4) GO TO 200
C
        CVAL=CDAT(LVAL+(IPARM-1)*2+1)
        IF(IP .NE. 6) GO TO 180
        ACT=CVAL
        EF=ACT*(1.-((B1*AWL/ACT)+B2))
        GO TO 190
C
180     I=KNSTR(IP,CVAL,ACT,EF,LPCT,LDAY,LTOUR,LBLK,IBLD)
C
        IF(I .NE. 0) GO TO 190
        ACT=ACT+1.
        EF=ACT*(1.-((B1*AWL/ACT)+B2))
        GO TO 180
190     IF(ACT .LE. CDAT(LBLK+ACBOFF)) GO TO 200
        CDAT(LBLK+ACBOFF)=ACT
        CDAT(LBLK+EFBOFF)=EF
        ICDAT(LBLK+CTBOFF)=IP
200     CONTINUE
220     CONTINUE
        GO TO 140
        END
```

SUBROUTINE MOVE

Subroutine MOVE is called to move N words from array S to array T. S and T frequently represent parts of larger arrays.

```
      SUBROUTINE MOVE(S,T,N)                                          2760
C                                                                     2761
C  MOVES N WORDS FROM ARRAY S TO ARRAY T                              2762
C                                                                     2763
      DIMENSION S(N),T(N)                                             2764
      IF(N .LE. 0) RETURN                                             2765
      DO 10 I=1,N                                                     2766
10    T(I)=S(I)                                                       2767
      RETURN                                                          2768
      END                                                             2769
```


SUBROUTINE MRGORD

Subroutine MRGORD (*merge order*) is called to set a new default output order from the qualifier of a READ or DISP command. Its arguments are arrays fitting the description of the ORDER parameter of subroutine GTDSPC. NEWORD represents the ordering of qualifier phrases in the last qualifier scanned. OLDORD represents an ordering of output phrases resulting from merging the "old" ordering with the "new" ordering.

Subroutine MOVE is called to move the contents of OLDORD to a temporary storage (TMPORD) where elements can be "erased" without affecting the original values in OLDORD. Elements of NEWORD are moved to OUTORD in their current order and the phrase types moved are erased from TMPORD. Any elements of OUTORD left unfilled by this process are filled by moving elements from TMPORD in the order in which they occur. Thus, a new DISP command output order is established.

```
      SUBROUTINE MRGORD(NEWORD,OLDORD,OUTORD)                         2770
C                                                                     2771
C  SETS OUTPUT ORDER FOR DISP COMMAND.    MERGES NEW INFORMATION      2772
C  INTO OLD TO ESTABLISH OUTPUT ORDER                                 2773
C                                                                     2774
      INTEGER OLDORD,OUTORD,TMPORD                                    2775
      DIMENSION NEWORD(3),OLDORD(3),OUTORD(3),TMPORD(3)               2776
C                                                                     2777
      CALL MOVE(OLDORD,TMPORD,3)                                      2778
      DO 30 IORD=1,3                                                  2779
      IF(NEWORD(IORD) .EQ. 0) GO TO 10                                2780
      OUTORD(IORD)=NEWORD(IORD)                                       2781
      I=LKP1(NEWORD(IORD),TMPORD,3)                                   2782
      IF(I .NE. 0) TMPORD(I)=0                                        2783
      GO TO 30                                                        2784
10    DO 20 I=1,3                                                     2785
      IF(TMPORD(I) .EQ. 0) GO TO 20                                   2786
      OUTORD(IORD)=TMPORD(I)                                          2787
      TMPORD(I)=0                                                     2788
      GO TO 30                                                        2789
20    CONTINUE                                                        2790
      RETURN                                                          2791
30    CONTINUE                                                        2792
      RETURN                                                          2793
      END                                                             2794
```

FUNCTION NXDAY

Function NXDAY (*next day*) is used to index through the selected days for a precinct during command execution. Its arguments LPCT and LDAY are pointers to the data for a precinct and to the data for the last day selected for the precinct, respectively (see Sec. IV). The value of the function is a pointer to the next day selected after LDAY (zero if none). On entry, a value of zero for LDAY indicates that the first day selected for the precinct is to be located.

A day is selected if and only if the value of its corresponding work flag is nonzero (see the discussion of subroutine SETWFL).

```
      FUNCTION NXDAY(LPCT,LDAY)                                          2795
C                                                                        2796
C  FINDS THE NEXT DAY SELECTED IN PRECINCT LPCT AFTER LDAY               2797
C                                                                        2798
C                                                                        2799
      COMMON/STORE/TOP,BOT,RDBOT,MAXBOT,NWORDS,CDAT(6000),C2DAT(6000)    2800
      INTEGER TOP,BOT,RDBOT                                              2801
      DIMENSION ICDAT(6000),IC2DAT(6000)                                 2802
      EQUIVALENCE(ICDAT,CDAT),(IC2DAT,C2DAT)                             2803
C                                                                        2804
C                                                                        2805
      COMMON/PNTRS/IOVRLY,IOVTR(2),                                      2806
     1NPCTDT,NPCTRD,LPCTDT,LNMLST(4),NNAMES(4),NDAYDT,LDAYNM,            2807
     2LDYRFL,NDAYRD,LDYWFL,NTRDT,LTRTB(2),LTRST,LTREND,LTRRFL,LTRNM,     2808
     3NTRRD,LTRWFL,NBLDT,LBLKTB(2),LBLRFL,NBLRD,LBLWFL,NDIVDT,NDIVRD,    2809
     4LDIVNM,LDIVFL                                                      2810
C                                                                        2811
      COMMON/OFFSET/NMPOFF,DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,    2812
     1NWDPCT,CPDOFF,SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,NWDDY,            2813
     2QDTOFF,QXTOFF,CRTOFF,QOTOFF,QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,    2814
     3PVTOFF,HFTOFF,MFTOFF,LFTOFF,NPRIO,NWDTR,BLDOFF,QOBOFF,QNBOFF,      2815
     4EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,NWDBL             2816
C                                                                        2817
      INTEGER DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,CPDOFF,          2818
     1SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,QDTOFF,QXTOFF,CRTOFF,QOTOFF,    2819
     2QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,PVTOFF,HFTOFF,BLDOFF,           2820
     3EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,QOBOFF,QNBOFF     2821
C                                                                        2822
      NXDAY=LDAY                                                         2823
      IF(LDAY .NE. 0) GO TO 10                                           2824
      NXDAY=LPCT+DYPOFF                                                  2825
      GO TO 20                                                           2826
10    NXDAY=NXDAY+NWDDY                                                  2827
20    IDAY=(NXDAY-LPCT-DYPOFF)/NWDDY+1                                   2828
      IF(IDAY .LE. NDAYRD) GO TO 30                                      2829
      NXDAY=0                                                            2830
```



```
      RETURN
30    IF(ICDAT(LDYWFL+IDAY-1) .EQ. 0) GO TO 10                           2831
      RETURN                                                             2832
      END                                                                2833
                                                                         2834
```

SUBROUTINE MOVE

Subroutine MOVE is called to move N words from array S to array T. S and T frequently represent parts of larger arrays.

```
      SUBROUTINE MOVE(S,T,N)                                          2760
C                                                                     2761
C  MOVES N WORDS FROM ARRAY S TO ARRAY T                              2762
C                                                                     2763
      DIMENSION S(N),T(N)                                             2764
      IF(N .LE. 0) RETURN                                             2765
      DO 10 I=1,N                                                     2766
10    T(I)=S(I)                                                       2767
      RETURN                                                          2768
      END                                                             2769
```


SUBROUTINE MRGORD

Subroutine MRGORD (*merge order*) is called to set a new default output order from the qualifier of a READ or DISP command. Its arguments are arrays fitting the description of the ORDER parameter of subroutine GTDSPC. NEWORD represents the ordering of qualifier phrases in the last qualifier scanned. OLDORD represents an ordering of output phrases resulting from merging the "old" ordering with the "new" ordering.

Subroutine MOVE is called to move the contents of OLDORD to a temporary storage (TMPORD) where elements can be "erased" without affecting the original values in OLDORD. Elements of NEWORD are moved to OUTORD in their current order and the phrase types moved are erased from TMPORD. Any elements of OUTORD left unfilled by this process are filled by moving elements from TMPORD in the order in which they occur. Thus, a new DISP command output order is established.

```
      SUBROUTINE MRGORD(NEWORD,OLDORD,OUTORD)                         2770
C                                                                     2771
C  SETS OUTPUT ORDER FOR DISP COMMAND.   MERGES NEW INFORMATION       2772
C  INTO OLD TO ESTABLISH OUTPUT ORDER                                 2773
C                                                                     2774
      INTEGER OLDORD,OUTORD,TMPORD                                    2775
      DIMENSION NEWORD(3),OLDORD(3),OUTORD(3),TMPORD(3)               2776
C                                                                     2777
      CALL MOVE(OLDORD,TMPORD,3)                                      2778
      DO 30 IORD=1,3                                                  2779
      IF(NEWORD(IORD) .EQ. 0) GO TO 10                                2780
      OUTORD(IORD)=NEWORD(IORD)                                       2781
      I=LKP1(NEWORD(IORD),TMPORD,3)                                   2782
      IF(I .NE. 0) TMPORD(I)=0                                        2783
      GO TO 30                                                        2784
10    DO 20 I=1,3                                                     2785
      IF(TMPORD(I) .EQ. 0) GO TO 20                                   2786
      OUTORD(IORD)=TMPORD(I)                                          2787
      TMPORD(I)=0                                                     2788
      GO TO 30                                                        2789
20    CONTINUE                                                        2790
      RETURN                                                          2791
30    CONTINUE                                                        2792
      RETURN                                                          2793
      END                                                             2794
```

FUNCTION NXDAY

Function NXDAY (*next day*) is used to index through the selected days for a precinct during command execution. Its arguments LPCT and LDAY are pointers to the data for a precinct and to the data for the last day selected for the precinct, respectively (see Sec. IV). The value of the function is a pointer to the next day selected after LDAY (zero if none). On entry, a value of zero for LDAY indicates that the first day selected for the precinct is to be located.

A day is selected if and only if the value of its corresponding work flag is nonzero (see the discussion of subroutine SETWFL).

```
      FUNCTION NXDAY(LPCT,LDAY)                                          2795
C                                                                        2796
C  FINDS THE NEXT DAY SELECTED IN PRECINCT LPCT AFTER LDAY               2797
C                                                                        2798
C                                                                        2799
      COMMON/STORE/TOP,BOT,RDBOT,MAXBOT,NWORDS,CDAT(6000),C2DAT(6000)    2800
      INTEGER TOP,BOT,RDBOT                                              2801
      DIMENSION ICDAT(6000),IC2DAT(6000)                                 2802
      EQUIVALENCE(ICDAT,CDAT),(IC2DAT,C2DAT)                             2803
C                                                                        2804
C                                                                        2805
      COMMON/PNTRS/IOVRLY,IOVTR(2),                                      2806
     1NPCTDT,NPCTRD,LPCTDT,LNMLST(4),NNAMES(4),NDAYDT,LDAYNM,            2807
     2LDYRFL,NDAYRD,LDYWFL,NTRDT,LTRTB(2),LTRST,LTREND,LTRRFL,LTRNM,     2808
     3NTRRD,LTRWFL,NBLDT,LBLKTB(2),LBLRFL,NBLRD,LBLWFL,NDIVDT,NDIVRD,    2809
     4LDIVNM,LDIVFL                                                      2810
C                                                                        2811
      COMMON/OFFSET/NMPOFF,DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,    2812
     1NWDPCT,CPDOFF,SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,NWDDY,            2813
     2QDTOFF,QXTOFF,CRTOFF,QOTOFF,QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,    2814
     3PVTOFF,HFTOFF,MFTOFF,LFTOFF,NPRIO,NWDTR,BLDOFF,QOBOFF,QNBOFF,      2815
     4EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,NWDBL             2816
C                                                                        2817
      INTEGER DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,CPDOFF,          2818
     1SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,QDTOFF,QXTOFF,CRTOFF,QOTOFF,    2819
     2QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,PVTOFF,HFTOFF,BLDOFF,           2820
     3EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,QOBOFF,QNBOFF     2821
C                                                                        2822
      NXDAY=LDAY                                                         2823
      IF(LDAY .NE. 0) GO TO 10                                           2824
      NXDAY=LPCT+DYPOFF                                                  2825
      GO TO 20                                                           2826
10    NXDAY=NXDAY+NWDDY                                                  2827
20    IDAY=(NXDAY-LPCT-DYPOFF)/NWDDY+1                                   2828
      IF(IDAY .LE. NDAYRD) GO TO 30                                      2829
      NXDAY=0                                                            2830
```



```
      RETURN                                                             2831
30    IF(ICDAT(LDYWFL+IDAY-1) .EQ. 0) GO TO 10                           2832
      RETURN                                                             2833
      END                                                                2834
```

FUNCTION NXPCT

Function NXPCT (*next precinct*) is called during command execution to determine the next precinct selected by a command qualifier. Its argument (LPCT) is a pointer to the data for a precinct in array CDAT. On entry, LPCT points to the last precinct selected (zero if none) and the value of the function is a pointer to the next precinct selected (zero if none).

A precinct is selected if any of the following criteria are met:

- No precinct or division names appear in the command qualifier
- The entry in the table pointed to by LDIVFL (see Sec. IV) that corresponds to the precinct's division is nonzero
- The precinct's name appears in the PRECINCT phrase of the command qualifier

```
      FUNCTION NXPCT(LPCT)                                                  2835
C                                                                           2836
C  FINDS THE NEXT PRECINCT SELECTED AFTER LPCT                              2837
C                                                                           2838
C                                                                           2839
      COMMON/STORE/TOP,BOT,RDBOT,MAXBOT,NWORDS,CDAT(6000),C2DAT(6000)       2840
      INTEGER TOP,BOT,RDBOT                                                 2841
      DIMENSION ICDAT(6000),IC2DAT(6000)                                    2842
      EQUIVALENCE(ICDAT,CDAT),(IC2DAT,C2DAT)                                2843
C                                                                           2844
C                                                                           2845
      COMMON/OFFSET/NMPOFF,DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,       2846
     1NWDPCT,CPDOFF,SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,NWDDY,               2847
     2QDTOFF,QXTOFF,CRTOFF,QOTOFF,QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,       2848
     3PVTOFF,HFTOFF,MFTOFF,LFTOFF,NPRIO,NWDTR,BLDOFF,QOBOFF,QNBOFF,         2849
     4EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,NWDBL                2850
C                                                                           2851
      INTEGER DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,CPDOFF,             2852
     1SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,QDTOFF,QXTOFF,CRTOFF,QOTOFF,       2853
     2QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,PVTOFF,HFTOFF,BLDOFF,              2854
     3EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,QOBOFF,QNBOFF        2855
C                                                                           2856
      COMMON/PNTRS/IOVRLY,IOVTR(2),                                         2857
     1NPCTDT,NPCTRD,LPCTDT,LNMLST(4),NNAMES(4),NDAYDT,LDAYNM,               2858
     2LDYRFL,NDAYRD,LDYWFL,NTRDT,LTRTB(2),LTRST,LTREND,LTRRFL,LTRNM,        2859
     3NTRRD,LTRWFL,NBLDT,LBLKTB(2),LBLRFL,NBLRD,LBLWFL,NDIVDT,NDIVRD,       2860
     4LDIVNM,LDIVFL                                                         2861
C                                                                           2862
```



```
      NXPCT=LPCT                                                            2863
      IF(LPCT .NE. 0) GO TO 10                                              2864
      NXPCT=LPCTDT                                                          2865
      GO TO 20                                                              2866
10    NXPCT=NXPCT+NWDPCT                                                    2867
20    IF(NXPCT .LE. LPCTDT+(NPCTRD-1)*NWDPCT) GO TO 30                      2868
      NXPCT=0                                                               2869
      RETURN                                                                2870
30    IF(NNAMES(3)+NNAMES(4) .EQ. 0) RETURN                                 2871
      IF(NNAMES(3) .EQ.0) GO TO 40                                          2872
      IDIV=ICDAT(NXPCT+DVPOFF)                                              2873
      IF(ICDAT(LDIVFL+IDIV-1) .NE. 0) RETURN                                2874
40    IF(NNAMES(4) .EQ. 0) GO TO 10                                         2875
      I=LKP8(ICDAT(NXPCT+NMPOFF),ICDAT(LNMLST(4)),NNAMES(4))                2876
      IF(I .EQ. 0) GO TO 10                                                 2877
      RETURN                                                                2878
      END                                                                   2879
```

FUNCTION NXTOUR

Function NXTOUR (*next tour*) is used to index through the selected tours of a day during command execution. Its arguments LDAY and LTOUR are pointers to the data for a day within a precinct and the last tour selected within the day, respectively. The value of the function is a pointer to the next tour selected after LTOUR (zero if none). On entry, a value of zero for LTOUR indicates that the first tour selected for the day is to be located.

A tour is selected if and only if the value of its corresponding work flag is nonzero and the tour type is not equal to 1 (a tour type of 1 indicates that the tour holds a place that would be occupied by an overlay tour, but there is no overlay tour for the specified day).

On return, the parameter ITYPE is set to the value of the tour work flag. This value is the position of the tour relative to all the tours in the database and is useful for referencing the tables that contain tour starting and ending times and mappings of tours to blocks.

```
f       FUNCTION NXTOUR(LDAY,LTOUR,ITYPE)                                2880
C                                                                        2881
C  FINDS THE NEXT TOUR SELECTED IN LDAY AFTER LTOUR.                     2882
C  ITYPE IS THE VALUE OF THE TOUR WORK FLAG.                             2883
C                                                                        2884
C                                                                        2885
        COMMON/STORE/TOP,BOT,RDBOT,MAXBOT,NWORDS,CDAT(6000),C2DAT(6000)  2886
        INTEGER TOP,BOT,RDBOT                                            2887
        DIMENSION ICDAT(6000),IC2DAT(6000)                               2888
        EQUIVALENCE(ICDAT,CDAT),(IC2DAT,C2DAT)                           2889
C                                                                        2890
C                                                                        2891
        COMMON/OFFSET/NMPOFF,DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,  2892
     1NWDPCT,CPDOFF,SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,NWDDY,            2893
     2QDTOFF,QXTOFF,CRTOFF,QOTOFF,QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,    2894
     3PVTOFF,HFTOFF,MFTOFF,LFTOFF,NPRIO,NWDTR,BLDOFF,QOBOFF,QNBOFF,      2895
     4EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,NWDBL             2896
C                                                                        2897
        INTEGER DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,CPDOFF,        2898
     1SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,QDTOFF,QXTOFF,CRTOFF,QOTOFF,    2899
     2QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,PVTOFF,HFTOFF,BLDOFF,           2900
     3EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,QOBOFF,QNBOFF     2901
C                                                                        2902
        COMMON/PNTRS/IOVRLY,IOVTR(2),                                    2903
     1NPCTDT,NPCTRD,LPCTDT,LNMLST(4),NNAMES(4),NDAYDT,LDAYNM,            2904
```



```
     2LDYRFL,NDAYRD,LDYWFL,NTRDT,LTRTB(2),LTRST,LTREND,LTRRFL,LTRNM,     2905
     3NTRRD,LTRWFL,NBLDT,LBLKTB(2),LBLRFL,NBLRD,LBLWFL,NDIVDT,NDIVRD,   2906
     4LDIVNM,LDIVFL                                                     2907
C                                                                       2908
      NXTOUR=LTOUR                                                      2909
      IF(NXTOUR .NE. 0) GO TO 10                                        2910
      NXTOUR=LDAY+TRDOFF                                                2911
      GO TO 20                                                          2912
10    NXTOUR=NXTOUR+NWDTR                                               2913
20    ITOUR=(NXTOUR-LDAY-TRDOFF)/NWDTR+1                                2914
      IF(ITOUR .LE. NTRRD) GO TO 30                                     2915
      NXTOUR=0                                                          2916
      RETURN                                                            2917
30    ITYPE=ICDAT(LTRWFL+ITOUR-1)                                       2918
      IF(ITYPE .EQ. 0 .OR. ICDAT(NXTOUR+TYTOFF) .EQ. 1) GO TO 10        2919
      RETURN                                                            2920
      END                                                               2921
```

FUNCTION OBJFUN

Function OBJFUN (*obj*ective *fun*ction) evaluates an objective function over a span of hours of a day. Parameter LPARM is a pointer to a number list that specifies the objective function to be evaluated and any associated parameters. ISTART and IEND specify the span of hours over which the function is to be evaluated. LPCT, LDAY, and LTOUR are pointers to the precinct, day, and tour in which the span of hours occurs. EF is the number of effective cars for which the function is to be evaluated.

OBJFUN selects the correct function subprogram to evaluate the objective function specified by LPARM. For objective functions 2 and 3, the second element of LPARM is the priority for which it is to be evaluated.

```
      FUNCTION OBJFUN(LPARM,ISTART,IEND,LPCT,LDAY,LTOUR,EF)                2922
C                                                                          2923
C  EVALUATES AN OBJECTIVE FUNCTION OVER THE SPAN OF HOURS                  2924
C  FROM ISTART TO IEND                                                     2925
C                                                                          2926
C                                                                          2927
      COMMON/STORE/TOP,BOT,RDBOT,MAXBOT,NWORDS,CDAT(6000),C2DAT(6000)      2928
      INTEGER TOP,BOT,RDBOT                                                2929
      DIMENSION ICDAT(6000),IC2DAT(6000)                                   2930
      EQUIVALENCE(ICDAT,CDAT),(IC2DAT,C2DAT)                               2931
C                                                                          2932
      DIMENSION C1(3),C2(3),C3(3)                                          2933
C                                                                          2934
      COMMON/OFFSET/NMPOFF,DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,      2935
     1NWDPCT,CPDOFF,SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,NWDDY,              2936
     2QDTOFF,QXTOFF,CRTOFF,QOTOFF,QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,      2937
     3PVTOFF,HFTOFF,MFTOFF,LFTOFF,NPRIO,NWDTR,BLDOFF,QOBOFF,QNBOFF,        2938
     4EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,NWDBL               2939
C                                                                          2940
      INTEGER DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,CPDOFF,            2941
     1SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,QDTOFF,QXTOFF,CRTOFF,QOTOFF,      2942
     2QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,PVTOFF,HFTOFF,BLDOFF,             2943
     3EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,QOBOFF,QNBOFF       2944
C                                                                          2945
      COMMON/PNTRS/IOVRLY,IOVTR(2),                                        2946
     1NPCTDT,NPCTRD,LPCTDT,LNMLST(4),NNAMES(4),NDAYDT,LDAYNM,              2947
     2LDYRFL,NDAYRD,LDYWFL,NTRDT,LTRTB(2),LTRST,LTREND,LTRRFL,LTRNM,       2948
     3NTRRD,LTRWFL,NBLDT,LBLKTB(2),LBLRFL,NBLRD,LBLWFL,NDIVDT,NDIVRD,      2949
     4LDIVNM,LDIVFL                                                       2950
C                                                                          2951
```



```
      IFNCTN=ICDAT(LPARM)                                                  2952
      LCR=LDAY+CRDOFF                                                      2953
      LST=LDAY+STDOFF                                                      2954
      DO 5 I=1,3                                                           2955
      C1(I)=C2DAT(LTOUR+QDTOFF+I-1)                                        2956
      C2(I)=C2DAT(LTOUR+QOTOFF+I-1)                                        2957
5     C3(I)=C2DAT(LTOUR+TYTOFF+I-1)                                        2958
      LFR=LTOUR+HFTOFF                                                     2959
      RV=CDAT(LTOUR+RVTOFF)                                                2960
      GO TO (10,20,30),IFNCTN                                              2961
C                                                                          2962
10    OBJFUN=OBJF1(0,ISTART,IEND,LPCT,LCR,LST,LFR,RV,EF,C1,C2,C3)          2963
      GO TO 100                                                            2964
20    N=ICDAT(LPARM+2)                                                     2965
      OBJFUN=OBJF2(N,ISTART,IEND,LPCT,LCR,LST,LFR,RV,EF,C1,C2,C3)          2966
      GO TO 100                                                            2967
30    RV=CDAT(LTOUR+RVTOFF)                                                2968
      OBJFUN=OBJF3(0,ISTART,IEND,LPCT,LCR,LST,LFR,RV,EF,C1,C2,C3)          2969
100   RETURN                                                               2970
      END                                                                  2971
```

FUNCTION OBJF1

Function OBJF1 (*obj*ective *f*unction *1*) returns the weighted sum of the probability that a call will be delayed over a span of hours of a day in precinct LPCT. It is called from COMPTB for DISP output, from KNSTR when constraints are being met, and from OBJFUN when car-hours are being allocated. Parameters ISTART and IEND specify the span of hours. LCR and LST are pointers to the hourly call-rate and service-time data for the day, EF is the number of effective cars on duty, and C1, C2, and C3 specify the dispatching policy for calls of priority 1, 2, and 3, respectively.

The probability that a call is delayed in each hour is obtained from subroutine TRIDSP; this is weighted by the number of calls in the hour. If the smoothing option is being used, different calculations are made for the first hour of the first tour, the hours within a block, and the first hour of a new block. The smoothing algorithm is described in App. A of the User's Manual.

```
      FUNCTION OBJF1(N,ISTART,IEND,LPCT,LCR,LST,LFR,RV,EF,C1,C2,C3)     2972
C                                                                       2973
C  CALCULATES WEIGHTED SUM OF PROBABILITY THAT A CALL                   2974
C  WILL BE DELAYED                                                      2975
C                                                                       2976
C                                                                       2977
      COMMON/STORE/TOP,BOT,RDBOT,MAXBOT,NWORDS,CDAT(6000),C2DAT(6000)   2978
      INTEGER TOP,BOT,RDBOT                                             2979
      DIMENSION ICDAT(6000),IC2DAT(6000)                                2980
      EQUIVALENCE(ICDAT,CDAT),(IC2DAT,C2DAT)                            2981
C                                                                       2982
      DIMENSION C1(3),C2(3),C3(3)                                       2983
C                                                                       2984
      COMMON/TRAVEL/SQRTA,STRDNF,SPEED,TRAVT,AVGTT                      2985
C                                                                       2986
      Q=0.0                                                             2987
      JSTART=ISTART                                                     2988
      PHP1=CDAT(LFR)                                                    2989
      PHP2=CDAT(LFR+1)                                                  2990
      ISMFLG=IC2DAT(1)                                                  2991
      IF(ISMFLG .EQ.0) GO TO 70                                         2992
      LPR=LCR                                                           2993
      LEFM1=LST                                                         2994
      EFM1=C2DAT(LEFM1+JSTART-1)                                        2995
      CR=CDAT(LCR+JSTART-1)                                             2996
      ST=CDAT(LST+JSTART-1)                                             2997
      IF(EFM1 .NE. -1.0)GO TO 7                                         2998
```



```
C                                                                       2999
C     FIRST HOUR OF FIRST TOUR READ                                     3000
C                                                                       3001
      PD1= TRIDSP(1,0,CR,ST,PHP1,PHP2,EF,C1,C2,C3,PI)                   3002
      Q=Q+PD1*CR                                                        3003
      C2DAT(LPR+JSTART-1)=PD1                                           3004
    5 IF (JSTART .EQ. IEND) GO TO 100                                   3005
      JSTART=JSTART+1                                                   3006
      GO TO 40                                                          3007
C                                                                       3008
C     CARRYOVER BETWEEN BLOCKS                                          3009
C                                                                       3010
    7 CRM1=CDAT(LCR+JSTART-2)                                           3011
      STM1=CDAT(LST+JSTART-2)                                           3012
      LPRIHR=LPR+JSTART                                                 3013
      PDM1=C2DAT(LPRIHR-2)                                              3014
      CR1=CR+CRLEFT(CRM1,STM1,PDM1,EFM1,PHP1,PHP2,C1,C2,C3)             3015
      PD1=PDEL(1,LPRIHR,CR,CR1,ST,PHP1,PHP2,EF,EFM1,C1,C2,C3)           3016
      Q=Q+PD1*CR                                                        3017
      GO TO 5                                                           3018
C                                                                       3019
C     SMOOTHING WITHIN A  BLOCK                                         3020
C                                                                       3021
   40 DO 50 IHOUR=JSTART,IEND                                           3022
      CR=CDAT(LCR+IHOUR-1)                                              3023
      ST=CDAT(LST+IHOUR-1)                                              3024
      LPRIHR=LPR+IHOUR                                                  3025
      PD1=PDEL(0,LPRIHR,CR,0,ST,PHP1,PHP2,EF,EFM1,C1,C2,C3)             3026
   50 Q=Q+PD1*CR                                                        3027
      GO TO 100                                                         3028
C                                                                       3029
C     NO SMOOTHING                                                      3030
C                                                                       3031
   70 DO 80 IHOUR=JSTART,IEND                                           3032
      CR=CDAT(LCR+IHOUR-1)                                              3033
      ST=CDAT(LST+IHOUR-1)                                              3034
      PD1=TRIDSP(1,0,CR,ST,PHP1,PHP2,EF,C1,C2,C3,PI)                    3035
   80 Q=Q+PD1*CR                                                        3036
  100 OBJF1=Q                                                           3037
      RETURN                                                            3038
       END                                                              3039
```

FUNCTION OBJF2

Function OBJF2 (*objective function 2*) returns the weighted sum of the average time that a call of a specified priority can expect to wait before dispatch. The sum is taken over a span of hours of a day in a precinct. The delay is in hours. Function OBJF2 also calculates the weighted sum of travel times to calls for service for use in function OBJF3.

Parameter N specifies the priority level of interest. ISTART and IEND specify the span of hours over which the weighted sum is to be taken. LST and LCR are pointers to the hourly service times and call rates for the day. LFR is a pointer to an array that contains the fraction of calls in each priority class for the shift in which the hours occur. EF is the number of effective cars on duty. C1, C2, and C3 specify the dispatching policy for calls of priority 1, 2, and 3, respectively. LPCT is a pointer to the data for the precinct and RV is the average speed that cars travel when responding to calls for service.

The formula used to compute the expected delay in an hour in a specified priority class is given in Ref. 2. The delay for each hour between ISTART and IEND is weighted by the number of calls in the priority class in the hour. If the smoothing option is being used, different calculations are made for the first hour of the first tour, the hours within a block and the first hour of a new block. The smoothing algorithm is described in App. A of the User's Manual.

```
      FUNCTION OBJF2(N,ISTART,IEND,LPCT,LCR,LST,LFR,RV,EF,C1,C2,C3)     3040
C                                                                       3041
C  COMPUTE WEIGHTED SUM OF PRIORITY N CALL DELAYS OVER                  3042
C  HOURS ISTART TO IEND                                                 3043
C                                                                       3044
      COMMON/STORE/TOP,BOT,RDBOT,MAXBOT,NWORDS,CDAT(6000),C2DAT(6000)   3045
      INTEGER TOP,BOT,RDBOT                                             3046
      DIMENSION ICDAT(6000),IC2DAT(6000)                                3047
      EQUIVALENCE(ICDAT,CDAT),(IC2DAT,C2DAT)                            3048
C                                                                       3049
      DIMENSION C1(3),C2(3),C3(3)                                       3050
C                                                                       3051
      COMMON/OFFSET/NMPOFF,DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,   3052
```



```
     1NWDPCT,CPDOFF,SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,NWDDY,           3053
     2QDTOFF,QXTOFF,CRTOFF,QOTOFF,QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,   3054
     3PVTOFF,HFTOFF,MFTOFF,LFTOFF,NPRIO,NWDTR,BLDOFF,QOBOFF,QNBOFF,     3055
     4EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,NWDBL            3056
C                                                                       3057
      INTEGER DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,CPDOFF,         3058
     1SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,QDTOFF,QXTOFF,CRTOFF,QOTOFF,   3059
     2QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,PVTOFF,HFTOFF,BLDOFF,          3060
     3EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,QOBOFF,QNBOFF    3061
C                                                                       3062
      COMMON/SYSTEM/SYSIN,SYSOUT,IFILE,LIT                              3063
      INTEGER SYSIN,SYSOUT                                              3064
C                                                                       3065
      COMMON/TRAVEL/SQRTA,STRDNF,SPEED,TRAVT,AVGTT                      3066
      W=0.0                                                             3067
      CRCARR=0.0                                                        3068
      AVGTT=0.0                                                         3069
      SPEED=RV                                                          3070
      A=CDAT(LPCT+ARPOFF)                                               3071
      SQRTA=SQRT(A)                                                     3072
      STRDNS=CDAT(LPCT+SMPOFF)/A                                        3073
      STRDNF=(STRDNS-1.0)/(STRDNS-2.0)                                  3074
C                                                                       3075
      JSTART=ISTART                                                     3076
      PHP1=CDAT(LFR)                                                    3077
      PHP2=CDAT(LFR+1)                                                  3078
      IF (N .EQ. 1) PHP=PHP1                                            3079
      IF (N .EQ. 2) PHP=PHP2                                            3080
      IF (N .EQ. 3) PHP=(1.0-PHP1-PHP2)                                 3081
      ISMFLG=IC2DAT(1)                                                  3082
      IF(N .LT. 1) GO TO 150                                            3083
      CMFRN1=0.                                                         3084
      NM1=N-1                                                           3085
      FRN=CDAT(LFR+NM1)                                                 3086
      IF (NM1 .LT. 1) GO TO 25                                          3087
      DO 15 I=1,NM1                                                     3088
   15 CMFRN1=CMFRN1+CDAT(LFR+I-1)                                       3089
   25 CMFRN=CMFRN1+FRN                                                  3090
      IF (ISMFLG .EQ. 0) GO TO 70                                       3091
      LPR=LCR                                                           3092
      LEFM1=LST                                                         3093
      EFM1=C2DAT(LEFM1+JSTART-1)                                        3094
      CR=CDAT(LCR+JSTART-1)                                             3095
      ST=CDAT(LST+JSTART-1)                                             3096
      IF(EFM1 .NE. -1.0) GO TO 7                                        3097
C                                                                       3098
C     FIRST HOUR OF FIRST TOUR READ                                     3099
C                                                                       3100
      W1=TRIDSP(2,N,CR,ST,PHP1,PHP2,EF,C1,C2,C3,PI)                     3101
      T1=TRAVT                                                          3102
      C2DAT(LPR+JSTART-1)=PI                                            3103
                                                                        3104
```

```
      W=W+W1*CR*PHP                                                      3105
      AVGTT=AVGTT+T1*CR*PHP                                              3106
    5 IF(JSTART. EQ. IEND) GO TO 100                                     3107
      JSTART=JSTART+1                                                    3108
      GO TO 40                                                           3109
C                                                                        3110
C     CARRYOVER BETWEEN BLOCKS                                           3111
C                                                                        3112
    7 W1=WLEFT(N,CR,ST,PHP1,PHP2,EF,EFM1,C1,C2,C3,LCR,LST,LPR,JSTART)    3113
      IF(W1 .LT. 6E+4) GO TO 8                                           3114
      W1=60.0                                                            3115
      CRCARR=CR                                                          3116
    8 T1=TRAVT                                                           3117
      W=W+W1*CR*PHP                                                      3118
      AVGTT=AVGTT+T1*CR*PHP                                              3119
      GO TO 5                                                            3120
C                                                                        3121
C     SMOOTHING WITHIN A BLOCK                                           3122
C                                                                        3123
   40 DO 51 IHOUR=JSTART, IEND                                           3124
      ICARFL=0                                                           3125
      CR=CDAT(LCR+IHOUR-1)                                               3126
      ST=CDAT(LST+IHOUR-1)                                               3127
      ENMU=EF/ST                                                         3128
      LPRIHR=LPR+IHOUR                                                   3129
   45 W1=TRIDSP(2,N,CR,ST,PHP1,PHP2,EF,C1,C2,C3,PI)                      3130
      IF(PI .LE. 0.99) GO TO 46                                          3131
      WSM=60.0                                                           3132
      ICARFL=1                                                           3133
      GO TO 49                                                           3134
   46 T1=TRAVT                                                           3135
      PD0=C2DAT(LPRIHR-2)                                                3136
      ZP=W1*PD0/PI*                                                      3137
     1   FRN/(ENMU*(1.0-CR*CMFRN/ENMU)*(1.0-CR*CMFRN1/ENMU))             3138
   48 WSM=0.5*W1+0.5*ZP                                                  3139
49    W=W+WSM*(CR+CRCARR)*PHP                                            3140
      CRCARR=0.0                                                         3141
      AVGTT=AVGTT+T1*CR*PHP                                              3142
   50 PD1=PDEL(0,LPRIHR,CR,0,ST,PHP1,PHP2,EF,EFM1,C1,C2,C3)              3143
      IF(ICARFL .EQ. 1) CRCARR=CRCARR+CR                                 3144
51    CONTINUE                                                           3145
C                                                                        3146
      GO TO 100                                                          3147
C                                                                        3148
C     NO SMOOTHING                                                       3149
C                                                                        3150
   70 DO 80 IHOUR=JSTART, IEND                                           3151
      CR=CDAT(LCR+IHOUR-1)                                               3152
      ST=CDAT(LST+IHOUR-1)                                               3153
      W1=TRIDSP(2,N,CR,ST,PHP1,PHP2,EF,C1,C2,C3,PI)                      3154
      T1=TRAVT                                                           3155
      W=W+W1*CR*PHP                                                      3156
```



```
80    AVGTT=AVGTT+T1*CR*PHP                                              3157
  100 OBJF2=W                                                            3158
      RETURN                                                             3159
C                                                                        3160
C     COMPUTE AVERAGE DISPATCH DELAY FOR ALL CALLS                       3161
C                                                                        3162
  150 IF (ISMFLG .EQ. 0) GO TO 700                                       3163
      LPR=LCR                                                            3164
      LEFM1=LST                                                          3165
      EFM1=C2DAT(LEFM1+JSTART-1)                                         3166
      CR=CDAT(LCR+JSTART-1)                                              3167
      ST=CDAT(LST+JSTART-1)                                              3168
      IF (EFM1 .NE. -1.0) GO TO 75                                       3169
C                                                                        3170
C     FIRST HOUR OF FIRST TOUR READ                                      3171
C                                                                        3172
      W1=TRIDSP(2,N,CR,ST,PHP1,PHP2,EF,C1,C2,C3,PI)                      3173
      T1=TRAVT                                                           3174
      C2DAT(LPR+JSTART-1)=PI                                             3175
      W=W+W1*CR                                                          3176
      AVGTT=AVGTT+T1*CR                                                  3177
   55 IF(JSTART. EQ. IEND) GO TO 1000                                    3178
      JSTART=JSTART+1                                                    3179
      GO TO 400                                                          3180
C                                                                        3181
C     FIRST HOUR CARRYOVER BETWEEN BLOCKS                                3182
C                                                                        3183
   75 W1=WLEFT(N,CR,ST,PHP1,PHP2,EF,EFM1,C1,C2,C3,LCR,LST,LPR,JSTART)    3184
      IF(W1 .LT. 6E+4) GO TO 78                                          3185
      W1=60.0                                                            3186
      CRCARR=CR                                                          3187
      T1=TRAVT                                                           3188
      W=W+W1*CR                                                          3189
78    AVGTT=AVGTT+T1*CR                                                  3190
      GO TO 55                                                           3191
C                                                                        3192
C     SMOOTHING WITHIN A BLOCK                                           3193
C                                                                        3194
  400 DO 501 IHOUR=JSTART,IEND                                           3195
      ICARFL=0                                                           3196
      CR=CDAT(LCR+IHOUR-1)                                               3197
      ST=CDAT(LST+IHOUR-1)                                               3198
      LPRIHR=LPR+IHOUR                                                   3199
      W1=TRIDSP(2,N,CR,ST,PHP1,PHP2,EF,C1,C2,C3,PI)                      3200
      T1=TRAVT                                                           3201
      PD0=C2DAT(LPRIHR-2)                                                3202
      IF(PI .GT. 0.99) GO TO 450                                         3203
      ZP=W1*PD0*PI                                                       3204
      GO TO 480                                                          3205
  450 WSM=60.0                                                           3206
      ICARFL=1                                                           3207
        GO TO 490                                                        3208
```

```
  480 WSM=0.5*W1+0.5*ZP                                                  3209
490   W=W+WSM*(CR+CRCARR)                                                3210
      CRCARR=0.0                                                         3211
      AVGTT=AVGTT+T1*CR                                                  3212
  500 PD1=PDEL(0,LPRIHR,CR,0,ST,PHP1,PHP2,EF,EFM1,C1,C2,C3)              3213
      IF(ICARFL .EQ. 1) CRCARR=CRCARR+CR                                 3214
501   CONTINUE                                                           3215
      GO TO 1000                                                         3216
C                                                                        3217
C     NO SMOOTHING                                                       3218
C                                                                        3219
  700 DO 800 IHOUR=JSTART,IEND                                           3220
      CR=CDAT(LCR+IHOUR-1)                                               3221
      ST=CDAT(LST+IHOUR-1)                                               3222
      W1=TRIDSP(2,N,CR,ST,PHP1,PHP2,EF,C1,C2,C3,PI)                      3223
      T1=TRAVT                                                           3224
      W=W+W1*CR                                                          3225
800   AVGTT=AVGTT+T1*CR                                                  3226
 1000 OBJF2=W                                                            3227
      RETURN                                                             3228
      END                                                                3229
```


FUNCTION OBJF3

Function OBJF3 (*objective function 3*) returns the weighted total delay (queuing + travel time) that a randomly selected call of a specified priority can expect to experience before a patrol car arrives, summed over a span of hours of a day in a precinct.

Parameter N specifies the priority level of interest. ISTART and IEND specify the span of hours over which the weighted sum is to be taken. LPCT is the pointer to the data for the precinct. RV is the response speed of patrol cars to calls for service, and EF is the number of effective cars on duty. LST and LCR are pointers to the hourly service times and call rates for the day. LFR is a pointer to an array that contains the fraction of calls in each priority class for the shift in which the hours occur. C1, C2, and C3 specify the dispatching policy for calls of priority 1, 2, and 3, respectively.

The formula for computing travel time is given in App. A of the User's Manual. The formula for computing the expected delay in an hour is given in Ref. 2. The queuing delays and travel times are weighted by the number of calls in each hour. They are calculated in function OBJF2. The weighted queuing time is returned as the value of the function. The weighted travel time is stored in AVGTT. If the smoothing option is being used, different calculations are made for the first hour of the first tour, the hours within a block and the first hour of a new block. The smoothing algorithm is described in App. A of the User's Manual.

```
      FUNCTION OBJF3(N,ISTART,IEND,LPCT,LCR,LST,LFR,RV,EF,C1,C2,C3)      3232
C                                                                        3233
C     OBTAINS  WEIGHTED SUM OF TOTAL RESPONSE TIMES                      3234
C                                                                        3235
      DIMENSION C1(3), C2(3), C3(3)                                      3236
C                                                                        3237
      COMMON/TRAVEL/SQRTA,STRDNF,SPEED,TRAVT,AVGTT                       3238
C                                                                        3239
      AVWAIT=OBJF2(N,ISTART,IEND,LPCT,LCR,LST,LFR,RV,EF,C1,C2,C3)        3240
      TOTDEL=AVWAIT+AVGTT                                                3241
      OBJF3=TOTDEL                                                       3242
C                                                                        3243
      RETURN                                                             3244
      END                                                                3245
```

FUNCTION PDEL

Function PDEL (*probability of delay*) is used to obtain the probability that a call will be delayed, given a call rate, service time, distribution of calls by priority, dispatch policy, and number of effective cars. It is envoked to obtain the probability of a call being queued before dispatch for one hour of a day in a precinct. The formula used to compute PDEL is given in Ref. 2.

PDEL is called only if the option of smoothing queuing behavior over time has been selected. If the current hour is not the first hour of a block, PDEL is the average of the (smoothed) delay probability in the previous hour (PDM1) and the (unsmoothed) delay probability in the current hour (PD). If this is the first hour of a block, then the call rate used in calculating the delay probability in the current hour includes the effect of calls already in queue at the end of the previous block. The call rate used in the calculation (CR1) is determined in subroutine CRLEFT.

```
      FUNCTION PDEL(ICARRY,LPRIHR,CR,CR1,ST,PHP1,PHP2,EF,EFM1,C1,C2,C3)  3246
C                                                                        3247
C     COMPUTE PROBABILITY OF DELAY FOR HOUR IHR                          3248
C                                                                        3249
      DIMENSION C1(3),C2(3),C3(3)                                        3250
C                                                                        3251
       COMMON/STORE/TOP,BOT,RDBOT,MAXBOT,NWORDS,CDAT(6000),C2DAT(6000)   3252
       INTEGER TOP,BOT,RDBOT                                             3253
       DIMENSION ICDAT(6000),IC2DAT(6000)                                3254
       EQUIVALENCE(ICDAT,CDAT),(IC2DAT,C2DAT)                            3255
C                                                                        3256
C                                                                        3257
C     SMOOTHING WITHIN BLOCK                                             3258
C                                                                        3259
      PD=TRIDSP(1,0,CR,ST,PHP1,PHP2,EF,C1,C2,C3,PI)                      3260
      PDM1=C2DAT(LPRIHR-2)                                               3261
      PDEL=0.5*PD+0.5*PDM1                                               3262
      C2DAT(LPRIHR-1)=PDEL                                               3263
      IF(ICARRY .EQ. 0) RETURN                                           3264
C                                                                        3265
C     FIRST HOUR CARRYOVER BETWEEN BLOCKS                                3266
C                                                                        3267
      IF(EF .EQ. EFM1) RETURN                                            3268
      PD1=TRIDSP(1,0,CR1,ST,PHP1,PHP2,EF,C1,C2,C3,PI)                    3269
      IF(EF .GT. EFM1) GO TO 10                                          3270
      PDEL=MAX(PD1,PDEL)                                                 3271
      GO TO 20                                                           3272
```



```
10 PDEL=MIN(PD1,PDEL)                                                    3273
20 C2DAT(LPRIHR-1)=PDEL                                                  3274
   RETURN                                                                3275
   END                                                                   3276
```

SUBROUTINE PRTBL

Subroutine PRTBL (*print table*) prints one line of DISP output. The line can represent any level of aggregation, from one shift to an overall average. Parameter ITAB specifies the table number to be printed[3] and that implies the format and number of items to be written. LEV is the level of aggregation of statistics that are to be printed. PRTBL assumes that T(LEV,N) contains the output measure that will be printed in column N+1 of the line of output (this will have been computed by TOTAL or COMPTB, depending on LEV). NAME is an eight-character identifier that will be printed in the first output column. In the current version NAME can be a tour name, a precinct name, or the word "AVERAGE," depending on the output order and the level of aggregation. FLAG is a one-character indicator that is printed at the left of an output line to show the overlay status of a shift.

```
      SUBROUTINE PRTBL(ITAB,LEV,FLAG,NAME)                              3277
C                                                                       3278
C  PRINTS ONE LINE OF TABLE ITAB                                        3279
C                                                                       3280
      COMMON/STATS/T(4,8),S(4,8),PORDER(3),RORDER(3),CIND(8)            3281
      INTEGER PORDER,RORDER                                             3282
C                                                                       3283
      COMMON/SYSTEM/SYSIN,SYSOUT,IFILE,LIT                              3284
      INTEGER SYSIN,SYSOUT                                              3285
C                                                                       3286
      DIMENSION NAME(8), IHOUR(6), IMIN(6), IMIN1(6), IMIN2(6)          3287
      INTEGER HOUR                                                      3288
       DATA BLANK/1H /                                                  3289
C                                                                       3290
      GO TO (100,200,300,400,500),ITAB                                  3291
C                                                                       3292
C ***************************************************************     3293
C                          TABLE 1                                  *   3294
C ***************************************************************     3295
C                                                                       3296
100   CONTINUE                                                          3297
      WRITE(SYSOUT,110) FLAG,NAME,(CIND(I),T(LEV,I),I=1,8)              3298
110   FORMAT(1H ,9A1,A1,F5.1,1X,A1,F5.1,5(2X,A1,F5.1),1X,A1,F5.1)       3299
      RETURN                                                            3300
C                                                                       3301
```

[3]This refers to the output reports that can be obtained using the DISP command, not to the tables in the present report.


```
C ***************************************************************     3302
C                          TABLE 2                                  *   3303
C ***************************************************************     3304
C                                                                       3305
200   CONTINUE                                                          3306
       FLAG=BLANK                                                       3307
      WRITE(SYSOUT,210) FLAG,NAME,(CIND(I),T(LEV,I),I=1,5)              3308
210   FORMAT(1H ,9A1,3X,A1,F5.1,4(4X,A1,F5.1))                          3309
      RETURN                                                            3310
C                                                                       3311
C ***************************************************************     3312
C                          TABLE 3                                  *   3313
C ***************************************************************     3314
C                                                                       3315
300   CONTINUE                                                          3316
C                                                                       3317
      J=0                                                               3318
      DO 305 I=3,7                                                      3319
      J=J+1                                                             3320
      IHOUR(J)=(T(LEV,I)+0.5)/60.0                                      3321
      IMIN(J)=MOD((T(LEV,I)+0.5),60.0)                                  3322
      X=FLOAT(IMIN(J))                                                  3323
      IMIN1(J)=X/10.0                                                   3324
      IMIN2(J)=MOD(X,10.0)                                              3325
305   CONTINUE                                                          3326
      IHOUR(6)=(T(LEV,8)+0.5)/60.0                                      3327
      IMIN(6)=MOD(T(LEV,8),60.0)                                        3328
      X=FLOAT(IMIN(6))                                                  3329
      IMIN1(6)=X/10.0                                                   3330
      IMIN2(6)=MOD(X,10.0)                                              3331
       ISEC=MOD((T(LEV,8)*60.0+0.5),60.0)                               3332
       X=FLOAT(ISEC)                                                    3333
       ISEC1=X/10.0                                                     3334
       ISEC2=MOD(X,10.0)                                                3335
      J=0                                                               3336
      DO 307 I=3,8                                                      3337
      J=J+1                                                             3338
      IF(T(LEV,I) .LT. 6E+4) GO TO 307                                  3339
      IHOUR(J)=99                                                       3340
      IMIN1(J)=9                                                        3341
      IMIN2(J)=9                                                        3342
      IF(I .NE. 8) GO TO 307                                            3343
      ISEC1=9                                                           3344
      ISEC2=9                                                           3345
307   CONTINUE                                                          3346
C                                                                       3347
C                                                                       3348
      WRITE(SYSOUT,310) FLAG,NAME,(CIND(I),T(LEV,I),I=1,2),             3349
     1 (CIND(J+2),IHOUR(J),IMIN1(J),IMIN2(J),J=1,6),ISEC1,ISEC2         3350
310   FORMAT(1H ,9A1,A1,F5.1,1X,A1,F5.1,6(1X,A1,I3,1H:,2I1),1H:,2I1)    3351
      RETURN                                                            3352
C                                                                       3353
```

```
C ***************************************************************
C                        TABLE 4                                *
C ***************************************************************
C
400     CONTINUE
        HOUR=T(LEV,7)
          MIN=MOD((T(LEV,7)*60.0+0.5),60.0)
        X=FLOAT(MIN)
        IMIN1(1)=X/10.0
        IMIN2(1)=MOD(X,10.0)
        IF(T(LEV,7) .LT. 6E+4) GO TO 409
        HOUR=99
        IMIN1(1)=9
        IMIN2(1)=9
409     CONTINUE
        WRITE(SYSOUT,410) FLAG,NAME,(CIND(I),T(LEV,I),I=1,6),
     1  CIND(7),HOUR,IMIN1(1),IMIN2(1)
410     FORMAT(1H ,9A1,1X,A1,F5.1,3X,A1,F5.1,2X,A1,F5.1,3(4X,A1,F5.1),
     1   3X,A1,I3,1H:,2I1)
        RETURN
C
C ***************************************************************
C                        TABLE 5                                *
C ***************************************************************
C
500     CONTINUE
C
        J=0
        DO 505 I=7,8
        J=J+1
        IHOUR(J)=(T(LEV,I)+0.5)/60.0
        IMIN(J)=MOD((T(LEV,I)+0.5),60.0)
        X=FLOAT(IMIN(J))
        IMIN1(J)=X/10.0
        IMIN2(J)=MOD(X,10.0)
        IF(T(LEV,I) .LT. 6E+4) GO TO 505
        IHOUR(J)=99
        IMIN1(J)=9
        IMIN2(J)=9
505     CONTINUE
C
        WRITE(SYSOUT,510) FLAG,NAME,(CIND(I),T(LEV,I),I=1,6),
     1  (CIND(J+6),IHOUR(J),IMIN1(J),IMIN2(J),J=1,2)
510     FORMAT(1H ,9A1,A1,F5.1,1X,A1,F5.1,1X,4(1X,A1,F5.1),1X,
     1             2(1X,A1,I3,1H:,2I1))
        RETURN
        END
```


SUBROUTINE READ

Subroutine READ carries out the READ command. Its function is to read selected data from the database and to make these data available to subsequent commands. See Sec. IV for a description of the organization of the data after they have been read.

READ calls GTDSPC to scan the command qualifier and MRGORD to set the default output order for subsequent DISP commands. Storage is then obtained for the various "work" flags (see Table 2, Sec. IV) and all "read" and "work" flags are initialized. If IOVRLY=1, a check is performed to insure that the overlaid tours have been specified.

When the program is reading data, precincts are selected on the basis of whether their precinct or division name appears in the qualifier. If no precinct or division names are specified, then all precincts are selected. Instead of storing a division name for each precinct read, the program uses a division number that refers to an entry in the table pointed to by LDIVNM (see Sec. IV).

Within selected precincts, days are selected on the basis of the value of entries in the table pointed to by LDYRFL (see Sec. IV). For each day, hourly call rates and service times are computed from the corresponding parameters and hourly factors; service times are converted from minutes to hours; and the first hour of the day is flagged for use in smoothing the queuing results over time.

Within days, tours are selected on the basis of the values of entries in the table pointed to by LTRRFL. A "tour type" is determined for each shift on the basis of its relationship to overlays (see Table 2). To facilitate indexing through the data, we have required that the same number of tours be stored for each day read, regardless of whether the day has an overlay tour. Therefore, the type of a tour is set to "ignore" when it occupies a position that would be held by an overlay tour but the database indicates that there is no overlay tour for the day. The meanings for other type codes should be apparent from Table 2.

Blocks are selected by entries in the table pointed to by LBLRFL. The constraint indicator for each block is set to -1 when it is read to indicate that it has not been in the scope of a prescriptive command (MEET, ADD, or ALOC).

Users have the option of having their input data checked for detectable errors. If error-checking has been requested, the READ subroutine makes sure that the incoming data are within bounds and are internally consistent. If errors are detected, an appropriate message is printed and the program is terminated after all the data have been read.

When the data for all tours and blocks of a day have been read, subroutine DERIVE is called to determine the numbers of actual and effective cars on duty in each block and to insure that enough cars are available in each block of the day to handle the cfs workload.

```
      SUBROUTINE READ                                                    3401
C                                                                        3402
C SUBROUTINE TO READ SELECTED DATA FROM DATA BASE                        3403
C                                                                        3404
      COMMON/STATS/T(4,8),S(4,8),PORDER(3),RORDER(3),CIND(8)             3405
      INTEGER PORDER,RORDER                                              3406
C                                                                        3407
      COMMON/SYSTEM/SYSIN,SYSOUT,IFILE,LIT                               3408
      INTEGER SYSIN,SYSOUT                                               3409
C                                                                        3410
      COMMON/OFFSET/NMPOFF,DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,    3411
     1NWDPCT,CPDOFF,SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,NWDDY,            3412
     2QDTOFF,QXTOFF,CRTOFF,QOTOFF,QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,    3413
     3PVTOFF,HFTOFF,MFTOFF,LFTOFF,NPRIO,NWDTR,BLDOFF,QOBOFF,QNBOFF,      3414
     4EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,NWDBL             3415
C                                                                        3416
      INTEGER DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,CPDOFF,          3417
     1SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,QDTOFF,QXTOFF,CRTOFF,QOTOFF,    3418
     2QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,PVTOFF,HFTOFF,BLDOFF,           3419
     3EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,QOBOFF,QNBOFF     3420
C                                                                        3421
      COMMON/KEYWDS/NKYWD,NTYPES,TYPOFF(4),KEYWD(8,30),WDTYPE(30)        3422
      INTEGER TYPOFF,WDTYPE                                              3423
      DIMENSION PCLSNM(8),DCLSNM(8),TOURNM(8)                            3424
      EQUIVALENCE (PCLSNM,KEYWD(1,4)),(DCLSNM,KEYWD(1,3)),               3425
     1(TOURNM,KEYWD(1,2))                                                3426
C                                                                        3427
C                                                                        3428
C                                                                        3429
C                                                                        3430
C                                                                        3431
      COMMON/PNTRS/IOVRLY,IOVTR(2),                                      3432
     1NPCTDT,NPCTRD,LPCTDT,LNMLST(4),NNAMES(4),NDAYDT,LDAYNM,            3433
     2LDYRFL,NDAYRD,LDYWFL,NTRDT,LTRTB(2),LTRST,LTREND,LTRRFL,LTRNM,     3434
```



```
     3NTRRD,LTRWFL,NBLDT,LBLKTB(2),LBLRFL,NBLRD,LBLWFL,NDIVDT,NDIVRD,    3435
     4LDIVNM,LDIVFL                                                      3436
C                                                                        3437
C                                                                        3438
      COMMON/STORE/TOP,BOT,RDBOT,MAXBOT,NWORDS,CDAT(6000),C2DAT(6000)    3439
      INTEGER TOP,BOT,RDBOT                                              3440
      DIMENSION ICDAT(6000),IC2DAT(6000)                                 3441
      EQUIVALENCE(ICDAT,CDAT),(IC2DAT,C2DAT)                             3442
C                                                                        3443
C                                                                        3444
      COMMON/SCODES/SEND,CMD,NUMLST,NAMLST,FSPEC,DSPEC,DUM,ERR           3445
      INTEGER SEND,CMD,FSPEC,DSPEC,DUM,ERR                               3446
C                                                                        3447
      COMMON /OPTION/ NOERCK                                             3448
       COMMON/TITLES/RTITLE(60),DTITLE(60),RUNFLG,DSNFLG                 3449
       INTEGER RUNFLG,DSNFLG                                             3450
C                                                                        3451
C                                                                        3452
      INTEGER TYPE,VAL,ORDER,OVRLY                                       3453
      DIMENSION VAL(2),TPCTNM(8),TDIVNM(8),TPCTDT(20),TDATA(24),         3454
     1ORDER(4)                                                           3455
      EQUIVALENCE (TPCTNM,TPCTDT),(TDIVNM,TPCTDT(9)),                    3456
     1(TPCTDT,TDATA),(OVRLY,IOVRLY)                                      3457
C                                                                        3458
C         INITIALIZE                                                     3459
C                                                                        3460
      NPCTRQ=0                                                           3461
      BOT=RDBOT                                                          3462
      LGETB=BOT                                                          3463
      LGETT=TOP                                                          3464
      NDIVRQ=0                                                           3465
      NDIVRD=0                                                           3466
      NPCTRD=0                                                           3467
      NDAYRD=0                                                           3468
      NTRRD=0                                                            3469
      NBLRD=0                                                            3470
C                                                                        3471
C         INTERPRET QUALIFIERS                                           3472
C                                                                        3473
      TYPE=CMD                                                           3474
      CALL SCAN(TYPE,VAL)                                                3475
      CALL GTDSPC(TYPE,VAL,ORDER)                                        3476
      IF(TYPE .NE. ERR) GO TO 10                                         3477
      TOP=LGETT                                                          3478
      RETURN                                                             3479
C                                                                        3480
C         SET DEFAULT OUTPUT ORDER FOR DISP                              3481
C                                                                        3482
10    CALL MRGORD(ORDER,PORDER,RORDER)                                   3483
C                                                                        3484
C         GET STORAGE FOR WORK FLAGS.                                    3485
C         INITIALIZE FLAGS                                               3486
```

```
C                    DATA FOR DAYS                                          3487
C                                                                           3488
      IT=1                                                                  3489
      L=LNMLST(IT)                                                          3490
C                                                                           3491
      IREAD=1                                                               3492
C                                                                           3493
      N=NNAMES(IT)                                                          3494
      IF(N .NE. 0) GO TO 30                                                 3495
      CALL GETBOT(NDAYDT,LDYWFL)                                            3496
      DO 20 I=1,NDAYDT                                                      3497
      ICDAT(LDYWFL+I-1)=I                                                   3498
20    ICDAT(LDYRFL+I-1)=I                                                   3499
      NDAYRD=NDAYDT                                                         3500
      GO TO 100                                                             3501
30    CALL GETBOT(N,LDYWFL)                                                 3502
        DO 40 I = 1,NDAYDT                                                  3503
40        ICDAT(LDYRFL+I-1) = 0                                             3504
        DO 50 I=1,N                                                         3505
        J=LKP8(CDAT(L),CDAT(LDAYNM),NDAYDT)                                 3506
        IF(J .EQ. 0) GO TO 900                                              3507
        ICDAT(LDYRFL+J-1)=1                                                 3508
50        L=L+8                                                             3509
        NDAYRD=N                                                            3510
        IDAY=0                                                              3511
        DO 60 I=1,NDAYDT                                                    3512
        IF(ICDAT(LDYRFL+I-1) .EQ. 0) GO TO 60                               3513
        IDAY=IDAY+1                                                         3514
        ICDAT(LDYRFL+I-1)=IDAY                                              3515
        ICDAT(LDYWFL+IDAY-1)=I                                              3516
60        CONTINUE                                                          3517
C                                                                           3518
C                    DATA FOR TOURS                                         3519
C                                                                           3520
100       IT=2                                                              3521
      N=NNAMES(IT)                                                          3522
      IF(N .NE. 0) GO TO 120                                                3523
      CALL GETBOT(NTRDT,LTRWFL)                                             3524
      DO 110 I=1,NTRDT                                                      3525
      ICDAT(LTRWFL+I-1)=I                                                   3526
110       ICDAT(LTRRFL+I-1)=I                                               3527
      NTRRD=NTRDT                                                           3528
      GO TO 200                                                             3529
120       L=LNMLST(IT)                                                      3530
      DO 130 I=1,NTRDT                                                      3531
130       ICDAT(LTRRFL+I-1)=0                                               3532
      CALL GETBOT(N,LTRWFL)                                                 3533
      DO 140 I=1,N                                                          3534
      J=LKP8(CDAT(L),CDAT(LTRNM),NTRDT)                                     3535
      IF(J .EQ. 0) GO TO 900                                                3536
      ICDAT(LTRRFL+J-1)=1                                                   3537
140       L=L+8                                                             3538
```



```
        NTRRD=N                                                             3539
        ITOUR=0                                                             3540
        DO 150 ITYPE=1,NTRDT                                                3541
      IF(ICDAT(LTRRFL+ITYPE-1) .EQ. 0) GO TO 150                            3542
      ITOUR=ITOUR+1                                                         3543
      ICDAT(LTRRFL+ITYPE-1)=ITOUR                                           3544
      ICDAT(LTRWFL+ITOUR-1)=ITYPE                                           3545
150   CONTINUE                                                              3546
C                                                                           3547
C                    DATA FOR DIVISIONS                                     3548
C                                                                           3549
200   IT=3                                                                  3550
      N=NNAMES(IT)                                                          3551
      NDIVRQ=N                                                              3552
      IF(N .EQ. 0) GO TO 300                                                3553
      IF( N .GT. NDIVDT) GO TO 910                                          3554
      L=LNMLST(IT)                                                          3555
      CALL MOVE(CDAT(L),CDAT(LDIVNM),8*N)                                   3556
C                                                                           3557
300   IT=4                                                                  3558
      NPCTRQ=NNAMES(IT)                                                     3559
      IF(NPCTRQ .GT. NPCTDT) GO TO 910                                      3560
      LPCTNM=LNMLST(IT)                                                     3561
C                                                                           3562
      DO 305 I=1,NBLDT                                                      3563
305   ICDAT(LBLRFL+I-1)=0                                                   3564
      N=NTRDT                                                               3565
      IF(OVRLY .NE. 0) N=N-1                                                3566
      DO 315 I=1,N                                                          3567
      IF(ICDAT(LTRRFL+I-1) .EQ. 0) GO TO 315                                3568
      DO 310 J=1,2                                                          3569
      K=ICDAT(LTRTB(J)+I-1)                                                 3570
      IF(K .EQ. 0) GO TO 315                                                3571
      NBLRD=NBLRD+1                                                         3572
      ICDAT(LBLRFL+K-1)=NBLRD                                               3573
310   CONTINUE                                                              3574
315   CONTINUE                                                              3575
C                                                                           3576
C          CHECK OVERLAY TOURS                                              3577
C                                                                           3578
      IF(OVRLY .EQ. 0 .OR. ICDAT(LTRRFL+NTRDT-1) .EQ. 0) GO TO 340          3579
      IBLK1=ICDAT(LTRTB(1)+NTRDT-1)                                         3580
      IBLK2=ICDAT(LTRTB(2)+NTRDT-1)                                         3581
      I=LKP1(IBLK1,ICDAT(LTRTB(2)),NTRDT)                                   3582
      IF(I .NE. 0) GO TO 325                                                3583
      N1 = 1                                                                3584
320   WRITE(SYSOUT,9) N1,TOURNM                                             3585
      STOP                                                                  3586
325   IOVTR(1)=ICDAT(LTRRFL+I-1)                                            3587
      IF(IOVTR(1) .EQ. 0) GO TO 320                                         3588
      I=LKP1(IBLK2,ICDAT(LTRTB(1)),NTRDT)                                   3589
      IF(I .NE. 0) GO TO 335                                                3590
```

```
      N2 = 2                                                              3591
330   WRITE(SYSOUT,9) N2,TOURNM                                           3592
      STOP                                                                3593
335   IOVTR(2)=ICDAT(LTRRFL+I-1)                                          3594
      IF(IOVTR(2) .EQ. 0) GO TO 330                                       3595
C                                                                         3596
340   NWDDY=TRDOFF+NTRRD*NWDTR+NBLRD*NWDBL                                3597
      NWDPCT=DYPOFF+NDAYRD*NWDDY                                          3598
      NDIVRD=NDIVRQ                                                       3599
      BLDOFF=TRDOFF+NTRRD*NWDTR                                           3600
      REWIND IFILE                                                        3601
      CALL SKIP(IFILE,(1+(NDAYDT-1)/10+1+1+NTRDT+DSNFLG))                 3602
C                                                                         3603
C           READ PRECINCT HEADER RECORD                                   3604
C                                                                         3605
      DO 450 IPCT=1,NPCTDT                                                3606
      READ(IFILE,1) TPCTDT                                                3607
      IF(NPCTRQ+NDIVRQ .NE. 0) GO TO 350                                  3608
  344 IF (NDIVRD .EQ. 0) GO TO 346                                        3609
345   IDIV=LKP8(TDIVNM,CDAT(LDIVNM),NDIVRD)                               3610
      IF(IDIV .NE. 0) GO TO 370                                           3611
 346    IDIV=0                                                           3612
      NDIVRD=NDIVRD+1                                                     3613
      IF(NDIVRD .LE. NDIVDT) GO TO 347                                    3614
      WRITE(SYSOUT,8) DCLSNM                                              3615
      STOP                                                                3616
347   CALL MOVE(TDIVNM,CDAT(LDIVNM+(NDIVRD-1)*8),8)                       3617
      IDIV=NDIVRD                                                         3618
      GO TO 370                                                           3619
C                                                                         3620
350   IF(NDIVRQ .EQ. 0) GO TO 355                                         3621
      IDIV=LKP8(TDIVNM,CDAT(LDIVNM),NDIVRQ)                               3622
      IF(IDIV .NE. 0) GO TO 370                                           3623
355   IF(NPCTRQ .EQ. 0) GO TO 360                                         3624
      J=LKP8(TPCTNM,CDAT(LPCTNM),NPCTRQ)                                  3625
      IF(J .NE. 0) GO TO 344                                              3626
360   CALL SKIP(IFILE,NDAYDT*(4+NTRDT))                                   3627
      GO TO 450                                                           3628
370   NPCTRD=NPCTRD+1                                                     3629
      CALL GETBOT(DYPOFF,LPCT)                                            3630
      IF(NPCTRD .EQ. 1) LPCTDT=LPCT                                       3631
      CALL MOVE(TPCTNM,CDAT(LPCT+NMPOFF),8)                               3632
      ICDAT(LPCT+DVPOFF)=IDIV                                             3633
      CALL MOVE(TPCTDT(17),CDAT(LPCT+ARPOFF),4)                           3634
C                                                                         3635
C                              CHECK ERRORS IN B1, B2                     3636
C                               NOERCK IS FLAG FOR ERROR CHECKS           3637
C                               NOERCK=1  DON'T DO ERROR CHECKS           3638
C                                     =0 (DEFAULT) ERROR CHECKS           3639
C                                     =-1 ERROR ALREADY FOUND             3640
      IF (NOERCK .GT. 0) GO TO 1040                                       3641
      IF ((CDAT(LPCT+B1POFF) .GE. -0.999 .AND.                            3642
```



```
     1     CDAT(LPCT+B1POFF) .LE. 0.999) .AND.
     2    (CDAT(LPCT+B2POFF) .GE. 0.0 .AND.
     3     CDAT(LPCT+B2POFF) .LE. 0.999)) GO TO 1030                      3643
      WRITE (SYSOUT,5003) PCLSNM,(ICDAT(LPCT+NMPOFF-1+I),I=1,8)           3644
      NOERCK = -1                                                         3645
1030  CONTINUE                                                            3646
      A = CDAT(LPCT+ARPOFF)                                               3647
      STRDNS = CDAT(LPCT+SMPOFF) / A                                      3648
      IF (STRDNS .GT. 3.0) GO TO 1040                                     3649
      WRITE (SYSOUT,5004) PCLSNM,(ICDAT(LPCT+NMPOFF-1+I),I=1,8)           3650
1040  CONTINUE                                                            3651
C                                                                         3652
C                                                                         3653
C           READ DAY DETAIL RECORDS FOR THIS PRECINCT                     3654
      DO 440 IDAY=1,NDAYDT                                                3655
      IF(ICDAT(LDYRFL+IDAY-1) .NE. 0) GO TO 375                           3656
      CALL SKIP(IFILE,(4+NTRDT))                                          3657
           GO TO 440                                                      3658
375   CALL GETBOT(TRDOFF,LDAY)                                            3659
      LCR=LDAY+CRDOFF-1                                                   3660
      READ(IFILE,2) CDAT(LDAY+CPDOFF),CDAT(LDAY+SPDOFF),                  3661
     C ICDAT(LDAY+OVDOFF),(CDAT(LCR+I),I=1,48)                            3662
      CPARM=CDAT(LDAY+CPDOFF)                                             3663
      SPARM=CDAT(LDAY+SPDOFF)                                             3664
C                                                                         3665
C                                                                         3666
C           CALCULATE CALL RATES AND SERVICE TIMES                        3667
      DO 380 I=1,24                                                       3668
      I1=I-1                                                              3669
      CDAT(LDAY+CRDOFF+I1)=CDAT(LDAY+CRDOFF+I1)*CPARM                     3670
      CDAT(LDAY+STDOFF+I1)=CDAT(LDAY+STDOFF+I1)*SPARM/60.                 3671
C                                                                         3672
C                                                                         3673
C                       ERROR CHECKS FOR                                  3674
C                        CALL RATES AND SERVICE TIME                      3675
                                                                          3676
      IF (NOERCK .GT. 0) GO TO 380                                        3677
      IF (CDAT(LDAY+CRDOFF+I1) .GT. 0.0) GO TO 1000                       3678
      JDAY=(LDAY-DYPOFF-LPCT)/NWDDY                                       3679
      JDAY=ICDAT(LDYWFL+JDAY)                                             3680
      LDNM=LDAYNM+(JDAY-1)*8-1                                            3681
      WRITE (SYSOUT,5000) PCLSNM,(ICDAT(LPCT+NMPOFF-1+K),K=1,8),          3682
     1    (ICDAT(LDNM+K),K=1,8)                                          3683
      NOERCK = -1                                                         3684
1000  IF (I1 .EQ. 0) GO TO 380                                            3685
      IF (CDAT(LDAY+STDOFF+I1) .GE. CDAT(LDAY+STDOFF)/3.0 .AND.           3686
     1    CDAT(LDAY+STDOFF+I1) .LE. CDAT(LDAY+STDOFF)*3.0)               3687
     2    GO TO 380                                                      3688
      JDAY=(LDAY-DYPOFF-LPCT)/NWDDY                                       3689
      JDAY=ICDAT(LDYWFL+JDAY)                                             3690
      LDNM=LDAYNM+(JDAY-1)*8-1                                            3691
      WRITE (SYSOUT,5001) PCLSNM,(ICDAT(LPCT+NMPOFF-1+K),K=1,8),          3692
     1    (ICDAT(LDNM+K),K=1,8)                                          3693
                                                                          3694
```

```
      NOERCK = -1
  380 CONTINUE
C
C          READ SHIFT DETAIL RECORDS
C          FOR THIS DAY AND PRECINCT
C
     JTOUR=0
      DO 400 ITOUR=1,NTRDT
      IF(ICDAT(LTRRFL+ITOUR-1) .NE. 0) GO TO 390
      CALL SKIP(IFILE,1)
      GO TO 400
  390 CALL GETBOT(NWDTR,LTOUR)
      JTOUR=JTOUR+1
      READ(IFILE,3) (CDAT(LTOUR+ACTOFF+I-1),I=1,5),
     1     C2DAT(LTOUR+QDTOFF+9),
     1    (C2DAT(LTOUR+QDTOFF+I-1),I=2,3),
     1    (C2DAT(LTOUR+QDTOFF+I-1),I=5,6),
     1    (C2DAT(LTOUR+QDTOFF+I-1),I=8,9)
      C2DAT(LTOUR+QDTOFF)=1.0E0-C2DAT(LTOUR+QDTOFF+1) -
     1 C2DAT(LTOUR+QDTOFF+2)
      C2DAT(LTOUR+QDTOFF+3)=1.0E0-C2DAT(LTOUR+QDTOFF+4) -
     1 C2DAT(LTOUR+QDTOFF+5)
      C2DAT(LTOUR+QDTOFF+6)=1.0E0-C2DAT(LTOUR+QDTOFF+7) -
     1 C2DAT(LTOUR+QDTOFF+8)
C
      IF(C2DAT(LTOUR+QDTOFF) .LT. 0.0 .OR.
     1    C2DAT(LTOUR+QOTOFF) .LT. 0.0 .OR.
     1    C2DAT(LTOUR+TYTOFF) .LT. 0.0    ) GO TO 391
      GO TO 392
C
  391 NOERCK=-1
      WRITE(SYSOUT,5006) PCLSNM,(ICDAT(LPCT+NMPOFF-1+I),I=1,8),
     1 (ICDAT(LDNM+I),I=1,8)
C
C                    ERROR CHECKS FOR
C                     CONSISTENT USE OF PRIORITIES
C
  392 IF (NOERCK .GT. 0) GO TO 1020
      IF (JTOUR .GT. 1) GO TO 1010
      MFLAG1 = 0
      MFLAG2 = 0
      MFLAG3 = 0
      NPRT = 0
      IF (CDAT(LTOUR+HFTOFF) .GT. 0.0) MFLAG1 = 1
      IF (CDAT(LTOUR+MFTOFF) .GT. 0.0) MFLAG2 = 1
      IF (CDAT(LTOUR+LFTOFF) .GT. 0.0) MFLAG3 = 1
      GO TO 1020
 1010 IF (((CDAT(LTOUR+HFTOFF) .EQ. 0.0 .AND. MFLAG1 .EQ. 0)
     1  .OR. (CDAT(LTOUR+HFTOFF) .GT. 0.0 .AND. MFLAG1 .EQ. 1))
     2  .AND.((CDAT(LTOUR+MFTOFF) .EQ. 0.0 .AND. MFLAG2 .EQ. 0)
     3  .OR. (CDAT(LTOUR+MFTOFF) .GT. 0.0 .AND. MFLAG2 .EQ. 1))
     4  .AND.((CDAT(LTOUR+LFTOFF) .EQ. 0.0 .AND. MFLAG3 .EQ. 0)
```



```
     5   .OR. (CDAT(LTOUR+LFTOFF) .GT. 0.0 .AND. MFLAG3 .EQ. 1)))
     6   GO TO 1020
      IF (NPRT .GT. 0) GO TO 1020
      JDAY=(LDAY-DYPOFF-LPCT)/NWDDY
      JDAY=ICDAT(LDYWFL+JDAY)
      LDNM=LDAYNM+(JDAY-1)*8-1
      WRITE (SYSOUT,5002) PCLSNM,(ICDAT(LPCT+NMPOFF-1+I),I=1,8),
     1     (ICDAT(LDNM+I),I=1,8)
      NPRT = 1
      NOERCK = -1
 1020 CONTINUE
      CDAT(LTOUR+QXTOFF) = -1.0
      ICDAT(LTOUR+TYTOFF) = 2
      IF(OVRLY .EQ. 0 .OR. ITOUR .LT. NTRDT) GO TO 400
      IF(ICDAT(LDAY+OVDOFF) .NE. 0) GO TO 395
      ICDAT(LTOUR+TYTOFF)=1
      GO TO 410
  395 ICDAT(LTOUR+TYTOFF)=5
      ICDAT(LDAY+TRDOFF+(IOVTR(1)-1)*NWDTR+TYTOFF)=3
      ICDAT(LDAY+TRDOFF+(IOVTR(2)-1)*NWDTR+TYTOFF)=4
      GO TO 410
  400 CONTINUE
C
C          READ BLANK DETAIL RECORD
C          FOR THIS DAY AND PRECINCT
  410 READ(IFILE,4) TDATA
C
  418 DO 420 I=1,NBLDT
       IF(ICDAT(LBLRFL+I-1) .EQ. 0) GO TO 420
       CALL GETBOT(NWDBL,LBLOCK)
       CDAT(LBLOCK+OCBOFF)=TDATA(I)
       ICDAT(LBLOCK+CTBOFF)=-1
  420 CONTINUE
C
C          CHECK THAT MINIMUM ALLOCATION IS PRESENT
C          AND CALCULATE AVERAGES
C
      IF (NOERCK .GE. 0) CALL DERIVE(LPCT,LDAY,IREAD)
  440 CONTINUE
  450 CONTINUE
      IF (NOERCK .LT. 0) STOP
      IF(NPCTRD .GT. 0) GO TO 460
      WRITE(SYSOUT,7) PCLSNM
      TOP=LGETT
      BOT=LGETB
      RETURN
  460 CALL GETBOT(NDIVRD,LDIVFL)
      TOP=LGETT
      RETURN
  900 WRITE(SYSOUT,5) (KEYWD(I,IT),I=1,8),(CDAT(L+I-1),I=1,8)
      GO TO 920
  910 WRITE(SYSOUT,6) (KEYWD(I,IT),I=1,8)
```

```
920     TOP=LGETT                                                       3799
        BOT=LGETB                                                       3800
        RETURN                                                          3801
C            FORMAT FOR PRECINCT HEADER RECORD                          3802
1       FORMAT(8A1,1X,8A1,1X,4(1X,F5.0))                                3803
C            FORMAT FOR DAY DETAIL RECORDS                              3804
2       FORMAT(2(F5.0,1X),I1/24F3.2/24F3.2)                             3805
C            FORMAT FOR SHIFT DETAIL RECORD                             3806
3       FORMAT(5(F5.0,1X),7(F5.0,1X))                                   3807
C            FORMAT FOR BLOCK DETAIL RECORD                             3808
4       FORMAT(24F3.1)                                                  3809
5       FORMAT(/' ***',2(1X,8A1),' NOT IN DATA - REENTER')              3810
6       FORMAT(/' *** TOO MANY ',8A1,'S SPECIFIED - REENTER')           3811
7       FORMAT(/' *** NO ',8A1,' DATA SELECTED - REENTER.')             3812
9       FORMAT(/' *** BLOCK ',I1,' FOR OVERLAY ',8A1,' NOT FOUND',      3813
    1' - EXECUTION TERMINATED')                                         3814
8       FORMAT(/' *** DATA BASE ERROR: MORE UNIQUE ',8A1,' NAMES THAN', 3815
    C  'DECLARED - EXECUTION TERMINATED')                               3816
 5000 FORMAT (/' *** CALL RATE IN',2(1X,8A1),' IS <= 0 FOR DAY ',       3817
    1    8A1,/'     PROGRAM WILL STOP AFTER ERROR CHECKS')              3818
 5001 FORMAT (/' *** SERVICE TIME PATTERN FOR',2(1X,8A1),' ON DAY ',    3819
    1    8A1,/'     IS PECULIAR.'                                       3820
    2   /'     PROGRAM WILL STOP AFTER ERROR CHECKS.')                  3821
 5002 FORMAT (/' *** INCONSISTENT USE OF PRIORITIES FOR',/              3822
    1    2(1X,8A1),' ON DAY ',8A1                                       3823
    2    /'     PROGRAM WILL STOP AFTER ERROR CHECKS.')                 3824
 5003 FORMAT (/' *** AN UNAVAILABILITY PARAMETER FOR',2(1X,8A1),/       3825
    1    ' IS OUT OF RANGE. THE PROPER RANGES ARE:',/                   3826
    2    '     -1<B1<1  AND  0<=B2<1',/                                 3827
    3    '     PROGRAM WILL STOP AFTER ERROR CHECKS.')                  3828
 5004 FORMAT (/' *** (STREET MILES)/AREA <= 3 FOR',2(1X,8A1),           3829
    1   /'     PROGRAM WILL STOP AFTER ERROR CHECKS.')                  3830
 5005 FORMAT (/' *** AVERAGE SUPPRESSIBLE CRIME FOR',2(1X,8A1),         3831
    1  ' IS <= 0 FOR DAY ',                                            3832
    2    8A1,/'     PROGRAM WILL STOP AFTER ERROR CHECKS')              3833
 5006 FORMAT (/' *** DISPATCH VALUE FOR PRIORITY 1,2,OR 3>0.0',/        3834
    1    2(1X,8A1),' ON DAY ',8A1                                       3835
    2    /'     PROGRAM WILL STOP AFTER ERROR CHECKS.')                 3836
      END                                                               3837
```


SUBROUTINE SBLACT

Subroutine SBLACT (*set block actual* cars) is called to determine the number of actual cars on duty in each block of a day in a precinct, based on the number of cars assigned to each tour of the day. Parameters LPCT and LDAY are pointers to the data for the precinct and day for which the block allocations are to be determined. The algorithm used to determine the number of cars on duty in each block of a day from the number of cars on duty in each tour of the day is given in Sec. III.

```
      SUBROUTINE SBLACT(LPCT,LDAY)                                      3838
C                                                                       3839
C  CALCULATES  NUMBER OF ACTUAL CARS IN BLOCKS,                         3840
C  BASED ON NUMBER OF CARS IN TOURS                                     3841
C                                                                       3842
      COMMON/STORE/TOP,BOT,RDBOT,MAXBOT,NWORDS,CDAT(6000),C2DAT(6000)   3843
      INTEGER TOP,BOT,RDBOT                                             3844
      DIMENSION ICDAT(6000),IC2DAT(6000)                                3845
      EQUIVALENCE(ICDAT,CDAT),(IC2DAT,C2DAT)                            3846
C                                                                       3847
C                                                                       3848
      COMMON/PNTRS/IOVRLY,IOVTR(2),                                     3849
     1NPCTDT,NPCTRD,LPCTDT,LNMLST(4),NNAMES(4),NDAYDT,LDAYNM,           3850
     2LDYRFL,NDAYRD,LDYWFL,NTRDT,LTRTB(2),LTRST,LTREND,LTRRFL,LTRNM,    3851
     3NTRRD,LTRWFL,NBLDT,LBLKTB(2),LBLRFL,NBLRD,LBLWFL,NDIVDT,NDIVRD,   3852
     4LDIVNM,LDIVFL                                                     3853
C                                                                       3854
      COMMON/OFFSET/NMPOFF,DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,   3855
     1NWDPCT,CPDOFF,SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,NWDDY,           3856
     2QDTOFF,QXTOFF,CRTOFF,QOTOFF,QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,   3857
     3PVTOFF,HFTOFF,MFTOFF,LFTOFF,NPRIO,NWDTR,BLDOFF,QOBOFF,QNBOFF,     3858
     4EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,NWDBL            3859
C                                                                       3860
      INTEGER DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,CPDOFF,         3861
     1SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,QDTOFF,QXTOFF,CRTOFF,QOTOFF,   3862
     2QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,PVTOFF,HFTOFF,BLDOFF,          3863
     3EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,QOBOFF,QNBOFF    3864
C                                                                       3865
      DO 10 IBLK=1,NBLRD                                                3866
      LBLK=LDAY+BLDOFF+(IBLK-1)*NWDBL                                   3867
10    CDAT(LBLK+ACBOFF)=0.                                              3868
      DO 30 ITYPE=1,NTRDT                                               3869
      ITOUR=ICDAT(LTRRFL+ITYPE-1)                                       3870
      IF(ITOUR .LT. 1) GO TO 30                                         3871
      LTOUR=LDAY+TRDOFF+(ITOUR-1)*NWDTR                                 3872
      IF(ICDAT(LTOUR+TYTOFF) .EQ. 1) GO TO 30                           3873
                                                                        3874
```

```
      DO 20 IB=1,2                                                              3875
      IBLKDT=ICDAT(LTRTB(IB)+ITYPE-1)                                           3876
      IF(IBLKDT .EQ. 0) GO TO 20                                                3877
      IBLKRD=ICDAT(LBLRFL+IBLKDT-1)                                             3878
      LBLK=LDAY+BLDOFF+(IBLKRD-1)*NWDBL                                         3879
      CDAT(LBLK+ACBOFF)=CDAT(LBLK+ACBOFF)+CDAT(LTOUR+ACTOFF)                    3880
      IF(ICDAT(LTOUR+TYTOFF) .EQ. 5) GO TO 20                                   3881
      DO 13 JJ=1,9                                                              3882
      C2DAT(LBLK+QDTOFF+JJ-1)=C2DAT(LTOUR+QDTOFF+JJ-1)                          3883
13    CONTINUE                                                                  3884
20    CONTINUE                                                                  3885
30    CONTINUE                                                                  3886
      RETURN                                                                    3887
      END                                                                       3888
```


SUBROUTINE SBLEF

Subroutine SBLEF (*set block effective cars*) determines the number of effective cars on duty in each block of a day. Parameters LPCT and LDAY are pointers to the data for the precinct and day for which the calculations are to be performed. Section II and App. A of the User's Manual give the formula used to compute effective cars from actual cars and average workload.

```
      SUBROUTINE SBLEF(LPCT,LDAY)                                               3889
C                                                                               3890
C  CONVERTS ACTUAL CARS TO EFFECTIVE CARS IN EACH BLOCK                         3891
C                                                                               3892
      COMMON/PNTRS/IOVRLY,IOVTR(2),                                             3893
     1NPCTDT,NPCTRD,LPCTDT,LNMLST(4),NNAMES(4),NDAYDT,LDAYNM,                   3894
     2LDYRFL,NDAYRD,LDYWFL,NTRDT,LTRTB(2),LTRST,LTREND,LTRRFL,LTRNM,            3895
     3NTRRD,LTRWFL,NBLDT,LBLKTB(2),LBLRFL,NBLRD,LBLWFL,NDIVDT,NDIVRD,           3896
     4LDIVNM,LDIVFL                                                             3897
C                                                                               3898
C                                                                               3899
      COMMON/STORE/TOP,BOT,RDBOT,MAXBOT,NWORDS,CDAT(6000),C2DAT(6000)           3900
      INTEGER TOP,BOT,RDBOT                                                     3901
      DIMENSION ICDAT(6000),IC2DAT(6000)                                        3902
      EQUIVALENCE(ICDAT,CDAT),(IC2DAT,C2DAT)                                    3903
C                                                                               3904
C                                                                               3905
      COMMON/OFFSET/NMPOFF,DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,           3906
     1NWDPCT,CPDOFF,SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,NWDDY,                   3907
     2QDTOFF,QXTOFF,CRTOFF,QOTOFF,QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,           3908
     3PVTOFF,HFTOFF,MFTOFF,LFTOFF,NPRIO,NWDTR,BLDOFF,QOBOFF,QNBOFF,             3909
     4EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,NWDBL                    3910
C                                                                               3911
      INTEGER DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,CPDOFF,                 3912
     1SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,QDTOFF,QXTOFF,CRTOFF,QOTOFF,           3913
     2QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,PVTOFF,HFTOFF,BLDOFF,                  3914
     3EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,QOBOFF,QNBOFF            3915
C                                                                               3916
      B1=CDAT(LPCT+B1POFF)                                                      3917
      B2=CDAT(LPCT+B2POFF)                                                      3918
      DO 10 IBLK=1,NBLRD                                                        3919
      LBLK=LDAY+BLDOFF+(IBLK-1)*NWDBL                                           3920
      AWL=CDAT(LBLK+AWBOFF)                                                     3921
C                                                                               3922
      ACT=CDAT(LBLK+ACBOFF)                                                     3923
10    CDAT(LBLK+EFBOFF)=ACT*(1.-((B1*AWL/ACT)+B2))                              3924
      RETURN                                                                    3925
      END                                                                       3926
```

SUBROUTINE SBLOBJ

Subroutine SBLOBJ increases the number of cars assigned to a block by one and evaluates a specified objective function with one car more than the new assignment. Both the new current objective function value and the objective function value with an additional car are saved.

LPARM is a pointer to the parameter list that specifies the objective function to be evaluated. LPCT, LDAY, LTOUR, and LBLK are pointers to the precinct, day, tour, and block to be operated upon. IBDT is the position of the type of block relative to blocks in the database.

```
      SUBROUTINE SBLOBJ(LPARM,LPCT,LDAY,LTOUR,LBLK,IBDT)                3927
C                                                                       3928
C DETERMINES THE OBJECTIVE FUNCTION FOR A BLOCK WITH ONE                3929
C MORE ACTUAL CAR THAN CURRENTLY ALLOCATED.                             3930
C                                                                       3931
      COMMON/PNTRS/IOVRLY,IOVTR(2),                                     3932
     1NPCTDT,NPCTRD,LPCTDT,LNMLST(4),NNAMES(4),NDAYDT,LDAYNM,           3933
     2LDYRFL,NDAYRD,LDYWFL,NTRDT,LTRTB(2),LTRST,LTREND,LTRRFL,LTRNM,    3934
     3NTRRD,LTRWFL,NBLDT,LBLKTB(2),LBLRFL,NBLRD,LBLWFL,NDIVDT,NDIVRD,   3935
     4LDIVNM,LDIVFL                                                     3936
C                                                                       3937
      COMMON/OFFSET/NMPOFF,DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,   3938
     1NWDPCT,CPDOFF,SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,NWDDY,           3939
     2QDTOFF,QXTOFF,CRTOFF,QOTOFF,QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,   3940
     3PVTOFF,HFTOFF,MFTOFF,LFTOFF,NPRIO,NWDTR,BLDOFF,QOBOFF,QNBOFF,     3941
     4EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,NWDBL            3942
C                                                                       3943
      INTEGER DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,CPDOFF,         3944
     1SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,QDTOFF,QXTOFF,CRTOFF,QOTOFF,   3945
     2QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,PVTOFF,HFTOFF,BLDOFF,          3946
     3EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,QOBOFF,QNBOFF    3947
C                                                                       3948
C                                                                       3949
      COMMON/STORE/TOP,BOT,RDBOT,MAXBOT,NWORDS,CDAT(6000),C2DAT(6000)   3950
      INTEGER TOP,BOT,RDBOT                                             3951
      DIMENSION ICDAT(6000),IC2DAT(6000)                                3952
      EQUIVALENCE(ICDAT,CDAT),(IC2DAT,C2DAT)                            3953
C                                                                       3954
C                                                                       3955
       CDAT(LBLK+QOBOFF)=CDAT(LBLK+QNBOFF)                              3956
       B1=CDAT(LPCT+B1POFF)                                             3957
       B2=CDAT(LPCT+B2POFF)                                             3958
       ISTART=ICDAT(LBLKTB(1)+IBDT-1)                                   3959
       IEND=ICDAT(LBLKTB(2)+IBDT-1)                                     3960
       ACT=CDAT(LBLK+ACBOFF)+1.                                         3961
       CDAT(LBLK+ACBOFF)=ACT                                            3962
```



```
       AWL=CDAT(LBLK+AWBOFF)
C                                                                       3963
                                                                        3964
       EF=ACT*(1.-((B1*AWL/ACT)+B2))                                    3965
       CDAT(LBLK+EFBOFF)=EF                                             3966
       ACT=ACT+1.                                                       3967
       EF=ACT*(1.-((B1*AWL/ACT)+B2))                                    3968
       CDAT(LBLK+QNBOFF)=OBJFUN(LPARM,ISTART,IEND,LPCT,LDAY,LTOUR,EF)   3969
       RETURN                                                           3970
       END                                                              3971
```

SUBROUTINE SCAN

This subroutine scans the user's command input for the next syntactic element (e.g., command identifier, name list, number list, etc.). Its two parameters STYPE and SVAL are set to the type and value, respectively, of the element obtained. SVAL is a two-word array; the meaning of each word depends on the type of the syntactic element, as shown in Table 10.

SCAN calls GETTKN to get the next lexical element (number, word, paren, etc.) from the command text. If STYPE indicates that the last element type was "end of command," then SCAN instructs GETTKN to start reading a new command by setting TYPE to indicate "end of command." (See description of subroutine GETTKN; the parameter is called LTYPE in SCAN.)

Table 10

SYNTACTIC TYPES RETURNED FROM SCAN

Type Identifier (STYPE)	Description	Form of SVAL
SEND	End of command encountered	--
CMD	Command identifier	(Position of identifier in KEYWD table, --)
NUMLST	Number list	(Number of elements in list, pointer to list)
NAMLST	Name list	(Number of elements in list, pointer to list)
FSPEC	Function identifier (objective function, constraint, data type, table)	(Position of identifier in KEYWD table, --)
DSPEC	Data type specification (DAY, TOUR, PRECINCT, DIVISION)	(Position of identifier in KEYWD table, --)
ERR	Invalid element	--

The elements of name lists are stored in the "top" of array CDAT in storage allocated by calls to GETTOP. Names are stored eight characters to a name, one character to a word. Elements of name lists occupy contiguous words of storage in the opposite order to the way they were entered.

Numbers are stored in word pairs. The first word to a pair contains the integer representation and the second word the floating point representation of the number. Word pairs in a number list occupy contiguous words of storage in the same order as they were entered.

```
      SUBROUTINE SCAN(STYPE,SVAL)                                          3972
C                                                                          3973
C  SCANS USER COMMAND INPUT FOR NEXT LEXICAL ELEMENT                       3974
C                                                                          3975
      COMMON/SYSTEM/SYSIN,SYSOUT,IFILE,LIT                                 3976
      INTEGER SYSIN,SYSOUT                                                 3977
C                                                                          3978
      COMMON/KEYWDS/NKYWD,NTYPES,TYPOFF(4),KEYWD(8,30),WDTYPE(30)          3979
      INTEGER TYPOFF,WDTYPE                                                3980
      DIMENSION PCLSNM(8),DCLSNM(8),TOURNM(8)                              3981
      EQUIVALENCE (PCLSNM,KEYWD(1,4)),(DCLSNM,KEYWD(1,3)),                 3982
     1(TOURNM,KEYWD(1,2))                                                  3983
C                                                                          3984
      COMMON/LCODES/LEND,WORD,NUM,LP,RP                                    3985
      INTEGER WORD,RP                                                      3986
C                                                                          3987
      COMMON/SCODES/SEND,CMD,NUMLST,NAMLST,FSPEC,DSPEC,DUM,ERR             3988
      INTEGER SEND,CMD,FSPEC,DSPEC,DUM,ERR                                 3989
C                                                                          3990
      COMMON/STORE/TOP,BOT,RDBOT,MAXBOT,NWORDS,CDAT(6000),C2DAT(6000)      3991
      INTEGER TOP,BOT,RDBOT                                                3992
      DIMENSION ICDAT(6000),IC2DAT(6000)                                   3993
      EQUIVALENCE(ICDAT,CDAT),(IC2DAT,C2DAT)                               3994
C                                                                          3995
C                                                                          3996
      INTEGER STYPE,SVAL                                                   3997
      DIMENSION SVAL(2),LVAL(8)                                            3998
C                                                                          3999
      LGETT=TOP                                                            4000
      IF(STYPE .EQ. SEND) LTYPE = LEND                                     4001
C                                                                          4002
C     GET NEXT LEXICAL ELEMENT FROM COMMAND                                4003
C                                                                          4004
10    CALL GETTKN(LTYPE,LVAL)                                              4005
      GO TO (100,200,300,400,405),LTYPE                                    4006
C                                                                          4007
                                                                           4008
```

```
C           END OF COMMAND REACHED                                    4009
C                                                                     4010
100     STYPE=SEND                                                    4011
        RETURN                                                        4012
C                                                                     4013
C           BEGINNING OF A WORD ENCOUNTERED                           4014
C                                                                     4015
200     I=LKP8(LVAL,KEYWD,NKYWD)                                      4016
        IF(I .EQ. 0) GO TO 220                                        4017
        STYPE=WDTYPE(I)                                               4018
        IF(STYPE .EQ. DUM) GO TO 10                                   4019
        SVAL(1)=I                                                     4020
        RETURN                                                        4021
C                                                                     4022
220     STYPE=NAMLST                                                  4023
        SVAL(1)=1                                                     4024
        CALL GETTOP(8,I)                                              4025
        CALL MOVE(LVAL,CDAT(I),8)                                     4026
        SVAL(2)=I                                                     4027
        RETURN                                                        4028
C                                                                     4029
C           NEXT LEXICAL ELEMENT IS A NUMBER                          4030
C                                                                     4031
300     STYPE=NUMLST                                                  4032
        SVAL(1)=1                                                     4033
        CALL GETTOP(2,I)                                              4034
        ICDAT(I)=LVAL(1)                                              4035
        ICDAT(I+1)=LVAL(2)                                            4036
        SVAL(2)=I                                                     4037
        RETURN                                                        4038
C                                                                     4039
C           NEXT LEXICAL ELEMENT IS A LEFT PARENTHESIS                4040
C                                                                     4041
400     CALL GETTKN(LTYPE,LVAL)                                       4042
        IF(LTYPE .EQ. NUM)GO TO 450                                   4043
        IF(LTYPE .EQ. WORD .OR. LTYPE .EQ. RP) GO TO 410              4044
C                                                                     4045
C           ERROR ENCOUNTERED IN COMMAND FORMAT                       4046
C                                                                     4047
405     WRITE(SYSOUT,1)                                               4048
1       FORMAT(/' **** INVALID LIST FORMAT - REENTER.')               4049
        STYPE=ERR                                                     4050
        RETURN                                                        4051
C                                                                     4052
C           NAMELIST ENCOUNTERED                                      4053
C                                                                     4054
410     N=0                                                           4055
        STYPE=NAMLST                                                  4056
415     IF(LTYPE .EQ. RP) GO TO 430                                   4057
        IF(LTYPE .EQ. WORD) GO TO 420                                 4058
        WRITE(SYSOUT,2)                                               4059
2       FORMAT(/' **** INVALID NAME LIST ELEMENT - REENTER.')         4060
```



```
        TOP=LGETT                                                     4061
        STYPE=ERR                                                     4062
        RETURN                                                        4063
420     N=N+1                                                         4064
        CALL GETTOP(8,LOC)                                            4065
        CALL MOVE(LVAL,CDAT(LOC),8)                                   4066
        CALL GETTKN(LTYPE,LVAL)                                       4067
        GO TO 415                                                     4068
430     SVAL(1)=N                                                     4069
        SVAL(2)=LOC                                                   4070
        RETURN                                                        4071
C                                                                     4072
C           NUMBERLIST ENCOUNTERED                                    4073
C                                                                     4074
450     N=0                                                           4075
        STYPE=NUMLST                                                  4076
460     IF(LTYPE .EQ. RP) GO TO 480                                   4077
        IF(LTYPE .EQ. NUM) GO TO 470                                  4078
        WRITE(SYSOUT,3)                                               4079
3       FORMAT(/' **** INVALID NUMBER LIST FORMAT - REENTER.')        4080
        STYPE=ERR                                                     4081
        TOP=LGETT                                                     4082
        RETURN                                                        4083
C                                                                     4084
C           STORE NUMBERS                                             4085
C                                                                     4086
470     N=N+1                                                         4087
        CALL GETTOP(2,LOC)                                            4088
        ICDAT(LOC)=LVAL(1)                                            4089
        ICDAT(LOC+1)=LVAL(2)                                          4090
        CALL GETTKN(LTYPE,LVAL)                                       4091
        GO TO 460                                                     4092
480     SVAL(1)=N                                                     4093
        SVAL(2)=LOC                                                   4094
        NSW=N/2                                                       4095
        IF(NSW .LT. 1) RETURN                                         4096
        J=LOC                                                         4097
        K=LOC+(N-1)*2                                                 4098
        DO 490 I=1,NSW                                                4099
        IT1=ICDAT(J)                                                  4100
        IT2=ICDAT(J+1)                                                4101
        ICDAT(J)=ICDAT(K)                                             4102
        ICDAT(J+1)=ICDAT(K+1)                                         4103
        ICDAT(K)=IT1                                                  4104
        ICDAT(K+1)=IT2                                                4105
        J=J+2                                                         4106
        K=K-2                                                         4107
490     CONTINUE                                                      4108
        RETURN                                                        4109
        END                                                           4110
```

SUBROUTINE SET

Subroutine SET carries out the SET command. Its function is to alter the values of specified data items that have been read from the database.

Successive calls to subroutine SCAN get pointers to lists of numbers that specify the types of data items to be altered and the values they are to assume. If the lists constitute a valid specification, subroutine GTDSPC is called to scan the command qualifier and SETWFL is called to set the "work" flags for the days and tours in the scope of the command.

In the main processing loop of SET, the program indexes through all selected precincts. For each precinct, the program indexes through all data item-value pairs specified by the user. If a data item applies to precincts as a whole (unavailability parameters are of this type), then the value of the data item for the precinct is changed to the specified value and the next data item-value pair is examined. If a data item applies to days within precincts (e.g., call-rate and service-time parameters) or to tours within days (e.g., actual cars assigned and response speed), then SET indexes through all selected days. If the data item applies to days as a whole, then the change is made for each day in turn. If the data item applies to tours, then the change is made to all selected tours within the day.

When all changes have been applied to all days for a precinct, subroutine DERIVE is called for each day to compute average workloads and effective cars for each block and to insure that the resulting number of effective cars on duty in each block of each day is sufficient to handle the cfs workload.

```
      SUBROUTINE SET                                                    4111
C                                                                       4112
C  IMPLEMENTS THE SET COMMAND                                           4113
C                                                                       4114
      COMMON/PNTRS/IOVRLY,IOVTR(2),                                     4115
     1NPCTDT,NPCTRD,LPCTDT,LNMLST(4),NNAMES(4),NDAYDT,LDAYNM,           4116
     2LDYRFL,NDAYRD,LDYWFL,NTRDT,LTRTB(2),LTRST,LTREND,LTRRFL,LTRNM,    4117
     3NTRRD,LTRWFL,NBLDT,LBLKTB(2),LBLRFL,NBLRD,LBLWFL,NDIVDT,NDIVRD,   4118
     4LDIVNM,LDIVFL                                                     4119
C                                                                       4120
```



```
      COMMON/SYSTEM/SYSIN,SYSOUT,IFILE,LIT                              4121
      INTEGER SYSIN,SYSOUT                                              4122
C                                                                       4123
C                                                                       4124
      COMMON/STORE/TOP,BOT,RDBOT,MAXBOT,NWORDS,CDAT(6000),C2DAT(6000)   4125
      INTEGER TOP,BOT,RDBOT                                             4126
      DIMENSION ICDAT(6000),IC2DAT(6000)                                4127
      EQUIVALENCE(ICDAT,CDAT),(IC2DAT,C2DAT)                            4128
C                                                                       4129
C                                                                       4130
      COMMON/KEYWDS/NKYWD,NTYPES,TYPOFF(4),KEYWD(8,30),WDTYPE(30)       4131
      INTEGER TYPOFF,WDTYPE                                             4132
      DIMENSION PCLSNM(8),DCLSNM(8),TOURNM(8)                           4133
      EQUIVALENCE (PCLSNM,KEYWD(1,4)),(DCLSNM,KEYWD(1,3)),              4134
     1(TOURNM,KEYWD(1,2))                                               4135
C                                                                       4136
      COMMON/OFFSET/NMPOFF,DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,   4137
     1NWDPCT,CPDOFF,SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,NWDDY,           4138
     2QDTOFF,QXTOFF,CRTOFF,QOTOFF,QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,   4139
     3PVTOFF,HFTOFF,MFTOFF,LFTOFF,NPRIO,NWDTR,BLDOFF,QOBOFF,QNBOFF,     4140
     4EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,NWDBL            4141
C                                                                       4142
      INTEGER DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,CPDOFF,         4143
     1SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,QDTOFF,QXTOFF,CRTOFF,QOTOFF,   4144
     2QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,PVTOFF,HFTOFF,BLDOFF,          4145
     3EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,QOBOFF,QNBOFF    4146
C                                                                       4147
      COMMON/SCODES/SEND,CMD,NUMLST,NAMLST,FSPEC,DSPEC,DUM,ERR          4148
      INTEGER SEND,CMD,FSPEC,DSPEC,DUM,ERR                              4149
C                                                                       4150
      DIMENSION ORDER(3),VAL(2)                                         4151
      INTEGER TYPE,VAL                                                  4152
C                                                                       4153
      IREAD=0                                                           4154
      LGETT=TOP                                                         4155
      TYPE=CMD                                                          4156
      CALL SCAN(TYPE,VAL)                                               4157
      IF(TYPE .EQ. FSPEC) GO TO 20                                      4158
10    WRITE(SYSOUT,1)                                                   4159
1     FORMAT(/' *** INVALID PARAMETER SPECIFICATION - REENTER')         4160
      TOP=LGETT                                                         4161
      RETURN                                                            4162
20    KEYVAL=VAL(1)                                                     4163
      I=KEYVAL-TYPOFF(FSPEC)                                            4164
      IF(I .NE. 1) GO TO 10                                             4165
C                                                                       4166
C     GET DATA TYPES, CHECK VALIDITY                                    4167
C                                                                       4168
      CALL SCAN(TYPE,VAL)                                               4169
      IF(TYPE .NE. NUMLST) GO TO 10                                     4170
      NPARM=VAL(1)                                                      4171
      LPARM=VAL(2)                                                      4172
```

```
      DO 25 IPARM=1,NPARM                                                       4173
      I=ICDAT(LPARM+(IPARM-1)*2)                                                4174
      IF(I .GT. 0 .AND. I .LT. 12) GO TO 25                                     4175
      WRITE(SYSOUT,4) I                                                         4176
4     FORMAT(/' *** PARAMETER ''',I2,''' INVALID - REENTER.')                   4177
      TOP=LGETT                                                                 4178
      RETURN                                                                    4179
25    CONTINUE                                                                  4180
C                                                                               4181
C           GET DATA VALUES                                                     4182
C                                                                               4183
      CALL SCAN(TYPE,VAL)                                                       4184
      IF(TYPE .EQ. NUMLST) GO TO 30                                             4185
      WRITE(SYSOUT,2)                                                           4186
2     FORMAT(/' *** INVALID PARAMETER VALUE - REENTER.')                        4187
      TOP=LGETT                                                                 4188
      RETURN                                                                    4189
30    NVAL=VAL(1)                                                               4190
      LVAL=VAL(2)                                                               4191
      IF(NVAL .EQ. NPARM) GO TO 40                                              4192
      WRITE(SYSOUT,3)                                                           4193
3     FORMAT(/' *** NUMBER OF VALUES DOES NOT MATCH NUMBER OF PARMS'            4194
     1,' - REENTER')                                                            4195
      TOP=LGETT                                                                 4196
      RETURN                                                                    4197
C                                                                               4198
C           SCAN QUALIFIER                                                      4199
C                                                                               4200
40    CALL SCAN(TYPE,VAL)                                                       4201
      CALL GTDSPC(TYPE,VAL,ORDER)                                               4202
      IF(TYPE .NE. ERR) GO TO 50                                                4203
      TOP=LGETT                                                                 4204
      RETURN                                                                    4205
C                                                                               4206
C           SET WORK FLAGS                                                      4207
C                                                                               4208
50    CALL SETWFL(IERR)                                                         4209
      IF(IERR .EQ. 0) GO TO 55                                                  4210
      TOP=LGETT                                                                 4211
      RETURN                                                                    4212
55    LPCT=0                                                                    4213
C                                                                               4214
C           GET NEXT PRECINCT                                                   4215
C                                                                               4216
60    LPCT=NXPCT(LPCT)                                                          4217
      IF(LPCT .EQ. 0) GO TO 140                                                 4218
C                                                                               4219
C           LOOK AT DATA TYPES                                                  4220
C                                                                               4221
      DO 130 IPARM=1,NPARM                                                      4222
      NP=ICDAT(LPARM+(IPARM-1)*2)                                               4223
C                                                                               4224
```



```
      IF(NP .EQ. 11) GO TO 120                                                  4225
C                                                                               4226
      IF(NP .GT. 2) GO TO 70                                                    4227
      CDAT(LPCT+SMPOFF+NP)=CDAT(LVAL+(IPARM-1)*2+1)                             4228
      GO TO 130                                                                 4229
C                                                                               4230
C           DAY-SPECIFIC DATA                                                   4231
C                                                                               4232
70    LDAY=0                                                                    4233
75    LDAY=NXDAY(LPCT,LDAY)                                                     4234
      IF(LDAY .EQ. 0) GO TO 130                                                 4235
      IF(NP .GT. 4) GO TO 90                                                    4236
      N=NP-3                                                                    4237
      XPARM=CDAT(LVAL+(IPARM-1)*2+1)                                            4238
      RATIO=XPARM/CDAT(LDAY+N)                                                  4239
      CDAT(LDAY+N)=XPARM                                                        4240
      L1=LDAY+CRDOFF                                                            4241
      IF(N.EQ.1) L1=LDAY+STDOFF                                                 4242
      L2=L1+23                                                                  4243
      DO 80 L=L1,L2                                                             4244
80    CDAT(L)=CDAT(L)*RATIO                                                     4245
      GO TO 75                                                                  4246
C                                                                               4247
C           TOUR-SPECIFIC DATA                                                  4248
C                                                                               4249
90    LTOUR=0                                                                   4250
95    LTOUR=NXTOUR(LDAY,LTOUR,ITYPE)                                            4251
      IF(LTOUR .EQ. 0) GO TO 75                                                 4252
97    IF(NP .EQ. 10) GO TO 100                                                  4253
      N=NP-5                                                                    4254
      CDAT(LTOUR+N+ACTOFF)=CDAT(LVAL+(IPARM-1)*2+1)                             4255
      GO TO 95                                                                  4256
C                                                                               4257
100   C2DAT(LTOUR+QDTOFF+9)=CDAT(LVAL+(IPARM-1)*2+1)                            4258
      GO TO 95                                                                  4259
C                                                                               4260
120   IC2DAT(1)=INT(CDAT(LVAL+(IPARM-1)*2+1))                                   4261
130   CONTINUE                                                                  4262
C                                                                               4263
C           RE-DERIVE BLOCK VALUES FOR EACH DAY AND CHECK FOR MINIMUM           4264
C                                                                               4265
      DO 135 IDAY=1,NDAYRD                                                      4266
      LDAY=LPCT+DYPOFF+(IDAY-1)*NWDDY                                           4267
135   CALL DERIVE(LPCT,LDAY,IREAD)                                              4268
      GO TO 60                                                                  4269
140   TOP=LGETT                                                                 4270
      RETURN                                                                    4271
      END                                                                       4272
```

SUBROUTINE SETWFL

Subroutine SETWFL (*set* work *fl*ags) is called to set the "work" flags (LTRWFL, LDYWFL) described in Table 2 after a command has been successfully interpreted. Division flags (LDIVFL) are also set.

These flags define the subsets of days, tours, and divisions (among those that have been read) that will be operated on by the current command. These subsets are determined from the phrases of the command qualifier. The qualifier phrases must have been converted to name lists by subroutine GTDSPC before SETWFL is called. For days and tours, each work flag corresponds to one day or tour that has been read. If a day or tour is selected by a command qualifier, then the value of its work flag will be the position of the day or tour relative to all the days or tours in the database; otherwise, its value will be zero. If no day or tour names appear in a command qualifier, then all days or tours read are implicitly selected; otherwise, only those names are selected. Names that do not appear in the database are ignored.

Division flags (in LDIVFL) correspond to names of divisions in a list produced by the command READ (LDIVNM). Flags of divisions named in command qualifier are set to one (1); others are set to zero (0). Division flags are referenced by a division number associated with each precinct. The condition of no division names in the command qualifier is detected in subroutine NXPCT.

The parameter IERR is set to 1 if any errors are detected in SETWFL; otherwise, its value on return will be zero.

```
      SUBROUTINE SETWFL(IERR)                                              4273
C                                                                          4274
C  SET WORK FLAGS                                                          4275
C BASED ON QUALIFIER SCANNED BY GTDSPC                                     4276
C                                                                          4277
      COMMON/SYSTEM/SYSIN,SYSOUT,IFILE,LIT                                 4278
      INTEGER SYSIN,SYSOUT                                                 4279
C                                                                          4280
      COMMON/KEYWDS/NKYWD,NTYPES,TYPOFF(4),KEYWD(8,30),WDTYPE(30)          4281
      INTEGER TYPOFF,WDTYPE                                                4282
      DIMENSION PCLSNM(8),DCLSNM(8),TOURNM(8)                              4283
      EQUIVALENCE (PCLSNM,KEYWD(1,4)),(DCLSNM,KEYWD(1,3)),                 4284
     1(TOURNM,KEYWD(1,2))                                                  4285
C                                                                          4286
C                                                                          4287
```



```
      COMMON/STORE/TOP,BOT,RDBOT,MAXBOT,NWORDS,CDAT(6000),C2DAT(6000)      4288
      INTEGER TOP,BOT,RDBOT                                                4289
      DIMENSION ICDAT(6000),IC2DAT(6000)                                   4290
      EQUIVALENCE(ICDAT,CDAT),(IC2DAT,C2DAT)                               4291
C                                                                          4292
C                                                                          4293
      COMMON/PNTRS/IOVRLY,IOVTR(2),                                        4294
     1NPCTDT,NPGTRD,LPCTDT,LNMLST(4),NNAMES(4),NDAYDT,LDAYNM,              4295
     2LDYRFL,NDAYRD,LDYWFL,NTRDT,LTRTB(2),LTRST,LTREND,LTRRFL,LTRNM,       4296
     3NTRRD,LTRWFL,NBLDT,LBLKTB(2),LBLRFL,NBLRD,LBLWFL,NDIVDT,NDIVRD,      4297
     4LDIVNM,LDIVFL                                                        4298
C                                                                          4299
C          SET DAY FLAGS                                                   4300
C                                                                          4301
      IERR=0                                                               4302
      IF(NDAYRD .LT. 1) GO TO 105                                          4303
      DO 10 I=1,NDAYRD                                                     4304
10    ICDAT(LDYWFL+I-1)=0                                                  4305
      LNM=LNMLST(1)                                                        4306
      N=NNAMES(1)                                                          4307
      IF(N.NE. 0) GO TO 30                                                 4308
      IDAY=0                                                               4309
      DO 20 I=1,NDAYDT                                                     4310
      IF(ICDAT(LDYRFL+I-1) .EQ. 0) GO TO 20                                4311
      IDAY=IDAY+1                                                          4312
      ICDAT(LDYWFL+IDAY-1)=I                                               4313
20    CONTINUE                                                             4314
      GO TO 100                                                            4315
30    IF(N .EQ. 0) GO TO 100                                               4316
      I=LKP8(ICDAT(LNM),ICDAT(LDAYNM),NDAYDT)                              4317
      IF(I .EQ. 0) GO TO 40                                                4318
      IDAY=ICDAT(LDYRFL+I-1)                                               4319
      IF(IDAY .LT. 1) GO TO 40                                             4320
      ICDAT(LDYWFL+IDAY-1)=I                                               4321
40    LNM=LNM+8                                                            4322
      N=N-1                                                                4323
      GO TO 30                                                             4324
C                                                                          4325
C          SET TOUR FLAGS                                                  4326
C                                                                          4327
100   IF(NTRRD .GT. 0) GO TO 108                                           4328
105   WRITE(SYSOUT,1)                                                      4329
1     FORMAT(/' *** NO PRIOR READ COMMAND - REENTER.')                     4330
      IERR=1                                                               4331
      RETURN                                                               4332
108   DO 110 I=1,NTRRD                                                     4333
110   ICDAT(LTRWFL+I-1)=0                                                  4334
      LNM=LNMLST(2)                                                        4335
      N=NNAMES(2)                                                          4336
      IF(N .NE. 0) GO TO 130                                               4337
      ITOUR=0                                                              4338
      DO 120 I=1,NTRDT                                                     4339
```

```
      IF(ICDAT(LTRRFL+I-1) .EQ. 0) GO TO 120                              4340
      ITOUR=ITOUR+1                                                       4341
      ICDAT(LTRWFL+ITOUR-1)=I                                             4342
120   CONTINUE                                                            4343
      GO TO 200                                                           4344
130   IF(N .EQ. 0) GO TO 200                                              4345
      I=LKP8(ICDAT(LNM),ICDAT(LTRNM),NTRDT)                               4346
      IF(I .EQ. 0) GO TO 140                                              4347
      ITOUR=ICDAT(LTRRFL+I-1)                                             4348
      IF(ITOUR .LT. 1) GO TO 140                                          4349
      ICDAT(LTRWFL+ITOUR-1)=I                                             4350
140   LNM=LNM+8                                                           4351
      N=N-1                                                               4352
      GO TO 130                                                           4353
C                                                                         4354
C          SET DIVISION FLAGS                                             4355
C                                                                         4356
200   IF(NDIVRD .EQ. 0) RETURN                                            4357
      DO 210 I=1,NDIVRD                                                   4358
210   ICDAT(LDIVFL+I-1)=0                                                 4359
      LNM=LNMLST(3)                                                       4360
      N=NNAMES(3)                                                         4361
220   IF(N .EQ. 0) RETURN                                                 4362
      I=LKP8(ICDAT(LNM),ICDAT(LDIVNM),NDIVRD)                             4363
      IF(I .EQ. 0) GO TO 230                                              4364
      ICDAT(LDIVFL+I-1)=1                                                 4365
230   LNM=LNM+8                                                           4366
      N=N-1                                                               4367
      GO TO 220                                                           4368
      END                                                                 4369
```


SUBROUTINE STRCAR

Subroutine STRCAR (*set tour cars*) determines a feasible allocation of cars to the tours of a day so that the resulting number of cars in each block of the day will be at least as great as the number currently assigned. The number of cars currently assigned to each block of the day is the number required to meet some constraint and is set by MEET or ADDALC. Parameter LDAY is a pointer to the data for the day for which the tour assignment is to be determined. CARHRS, on return, is the total number of car-hours that have been assigned to all tours of the day. The algorithm used to generate the assignment of cars to tours is given in Sec. III.

```
      SUBROUTINE STRCAR(LDAY,CARHRS)                                      4380
C                                                                         4381
C  DETERMINES FEASIBLE ALLOCATION OF CARS TO TOURS IN A                   4382
C DAY, GIVEN THE CAR REQUIREMENTS IN THE BLOCKS OF A DAY.                 4383
C                                                                         4384
C                                                                         4385
      COMMON/STORE/TOP,BOT,RDBOT,MAXBOT,NWORDS,CDAT(6000),C2DAT(6000)     4386
      INTEGER TOP,BOT,RDBOT                                               4387
      DIMENSION ICDAT(6000),IC2DAT(6000)                                  4388
      EQUIVALENCE(ICDAT,CDAT),(IC2DAT,C2DAT)                              4389
C                                                                         4390
C                                                                         4391
      COMMON/PNTRS/IOVRLY,IOVTR(2),                                       4392
     1NPCTDT,NPCTRD,LPCTDT,LNMLST(4),NNAMES(4),NDAYDT,LDAYNM,             4393
     2LDYRFL,NDAYRD,LDYWFL,NTRDT,LTRTB(2),LTRST,LTREND,LTRRFL,LTRNM,      4394
     3NTRRD,LTRWFL,NBLDT,LBLKTB(2),LBLRFL,NBLRD,LBLWFL,NDIVDT,NDIVRD,     4395
     4LDIVNM,LDIVFL                                                       4396
C                                                                         4397
      COMMON/OFFSET/NMPOFF,DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,     4398
     1NWDPCT,CPDOFF,SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,NWDDY,             4399
     2QDTOFF,QXTOFF,CRTOFF,QOTOFF,QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,     4400
     3PVTOFF,HFTOFF,MFTOFF,LFTOFF,NPRIO,NWDTR,BLDOFF,QOBOFF,QNBOFF,       4401
     4EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,NWDBL              4402
C                                                                         4403
      INTEGER DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,CPDOFF,           4404
     1SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,QDTOFF,QXTOFF,CRTOFF,QOTOFF,     4405
     2QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,PVTOFF,HFTOFF,BLDOFF,            4406
     3EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,QOBOFF,QNBOFF      4407
C                                                                         4408
      DIMENSION LBLK(2)                                                   4409
C                                                                         4410
      CARHRS=0.                                                           4411
      LTOUR=0                                                             4412
```

```
5       LTOUR=NXTOUR(LDAY,LTOUR,ITYPE)                                    4413
        IF(LTOUR .EQ. 0) RETURN                                           4414
        ISTART=ICDAT(LTRST+ITYPE-1)                                       4415
        IEND=ICDAT(LTREND+ITYPE-1)                                        4416
        TOURLN=IEND-ISTART+1                                              4417
C                                                                         4418
C            GET POINTERS TO BLOCKS                                       4419
C                                                                         4420
        DO 10 IB=1,2                                                      4421
        LBLK(IB)=0                                                        4422
        IBLK=ICDAT(LTRTB(IB)+ITYPE-1)                                     4423
        IF(IBLK .EQ. 0) GO TO 10                                          4424
        IBLK=ICDAT(LBLRFL+IBLK-1)                                         4425
        LBLK(IB)=LDAY+BLDOFF+(IBLK-1)*NWDBL                               4426
10       CONTINUE                                                         4427
        ID=ICDAT(LTOUR+TYTOFF)                                            4428
        GO TO (100,20,30,40,50),ID                                        4429
C                                                                         4430
C            TOUR NOT IN OVERLAY SEGMENT                                  4431
C                                                                         4432
20      I=1                                                               4433
        IF(LBLK(2) .EQ. 0) GO TO 25                                       4434
        IF(CDAT(LBLK(1)+ACBOFF) .LT. CDAT(LBLK(2)+ACBOFF)) I=2            4435
25      CDAT(LTOUR+ACTOFF)=CDAT(LBLK(I)+ACBOFF)                           4436
        CARHRS=CARHRS+TOURLN*CDAT(LTOUR+ACTOFF)                           4437
        ICDAT(LTOUR+CTTOFF)=ICDAT(LBLK(I)+CTBOFF)                         4438
        GO TO 5                                                           4439
C                                                                         4440
C            FIRST OVERLAID TOUR                                          4441
C                                                                         4442
30      X1=CDAT(LBLK(1)+ACBOFF)                                           4443
        CDAT(LTOUR+ACTOFF)=X1                                             4444
        CARHRS=CARHRS+TOURLN*X1                                           4445
        ICDAT(LTOUR+CTTOFF)=ICDAT(LBLK(1)+CTBOFF)                         4446
        X2=CDAT(LBLK(2)+ACBOFF)                                           4447
        GO TO 5                                                           4448
C                                                                         4449
C            SECOND OVERLAID TOUR                                         4450
C                                                                         4451
40      X3=CDAT(LBLK(1)+ACBOFF)                                           4452
        X4=CDAT(LBLK(2)+ACBOFF)                                           4453
        CDAT(LTOUR+ACTOFF)=X4                                             4454
        CARHRS=CARHRS+X4*TOURLN                                           4455
        ICDAT(LTOUR+CTTOFF)=ICDAT(LBLK(2)+CTBOFF)                         4456
        GO TO 5                                                           4457
C                                                                         4458
C            OVERLAY TOUR                                                 4459
C                                                                         4460
50      CDAT(LTOUR+ACTOFF)=AMAX1(X2-X1,X3-X4,0.)                          4461
        CARHRS=CARHRS+CDAT(LTOUR+ACTOFF)*TOURLN                           4462
        I=1                                                               4463
        IF(X3-X4 .GT. X2-X1) I=2                                          4464
```



```
        ICDAT(LTOUR+CTTOFF)=ICDAT(LBLK(I)+CTBOFF)                         4465
        IF(X2-X1 .LE. 0. .AND. X3-X4 .LE. 0.) ICDAT(LTOUR+CTTOFF)=0       4466
C                                                                         4467
C            ADJUST OVERLAY SEGMENT ASSIGNMENTS IF THE NUMBER OF CAR      4468
C            HOURS USED CAN BE REDUCED                                    4469
C                                                                         4470
        LOVTR=LTOUR                                                       4471
        ENOV=CDAT(LOVTR+ACTOFF)                                           4472
        IF(ENOV .EQ. 0.) RETURN                                           4473
        DELTA=AMAX1(X2-X1,0.)-AMAX1(X3-X4,0.)                             4474
        IF(DELTA .EQ. 0.) RETURN                                          4475
        ISW=1                                                             4476
        IF(DELTA .LT. 0.) ISW=2                                           4477
        DELTA=ABS(DELTA)                                                  4478
        ITRRD=IOVTR(ISW)                                                  4479
        ITYPE=ICDAT(LTRWFL+ITRRD-1)                                       4480
        ILEN=ICDAT(LTREND+ITYPE-1)-ICDAT(LTRST+ITYPE-1)+1                 4481
        IOVLN=ICDAT(LTREND+NTRDT-1)-ICDAT(LTRST+NTRDT-1)+1                4482
        IF(ILEN .GE. IOVLN) RETURN                                        4483
        LTOUR=LDAY+TRDOFF+(ITRRD-1)*NWDTR                                 4484
        LPCT=((LDAY-LPCTDT)/NWDPCT)*NWDPCT+LPCTDT                         4485
        B1=CDAT(LPCT+B1POFF)                                              4486
        B2=CDAT(LPCT+B2POFF)                                              4487
        CDAT(LOVTR+ACTOFF)=CDAT(LOVTR+ACTOFF)-DELTA                       4488
        CDAT(LTOUR+ACTOFF)=CDAT(LTOUR+ACTOFF)+DELTA                       4489
        CARHRS=CARHRS-DELTA*(IOVLN-ILEN)                                  4490
        IBDT=ICDAT(LTRTB(ISW)+ITYPE-1)                                    4491
        IBRD=ICDAT(LBLRFL+IBDT-1)                                         4492
        LBLOCK=LDAY+BLDOFF+(IBRD-1)*NWDBL                                 4493
        ACT=CDAT(LBLOCK+ACBOFF)+DELTA                                     4494
        CDAT(LBLOCK+ACBOFF)=ACT                                           4495
        AWL=CDAT(LBLOCK+AWBOFF)                                           4496
C                                                                         4497
        CDAT(LBLOCK+EFBOFF)=ACT*(1.-((B1*AWL/ACT)+B2))                    4498
        INV=2/ISW                                                         4499
        IBDT=ICDAT(LTRTB(INV)+NTRDT-1)                                    4500
        IBRD=ICDAT(LBLRFL+IBDT-1)                                         4501
        LBLOCK=LDAY+BLDOFF+(IBRD-1)*NWDBL                                 4502
        ACT=CDAT(LBLOCK+ACBOFF)-DELTA                                     4503
        CDAT(LBLOCK+ACBOFF)=ACT                                           4504
        AWL=CDAT(LBLOCK+AWBOFF)                                           4505
C                                                                         4506
        CDAT(LBLOCK+EFBOFF)=ACT*(1.-((B1*AWL/ACT)+B2))                    4507
100     RETURN                                                            4508
        END                                                               4509
```

SUBROUTINE STRDF

Subroutine STRDF determines the change in objective function value per car-hour that would be realized by making an incremental change in the assignment of cars to a shift. LDAY and LTOUR are pointers to the day and shift. ITYPE is the position of the tour relative to all tours in the database.

For any shift, a difference is computed by summing the contribution to the objective function of its blocks with the current number of cars and with one additional car assigned, and dividing by the length of the tour. For Type 1 overlay shifts an additional difference is obtained, summing the current and proposed objective function values of the first block of the first overlaid shift and the second block of the second overlaid shift, subtracting the sums, and dividing by the difference between the sum of the lengths of the overlaid tours and the length of the overlay tour.

```
      SUBROUTINE STRDF(LDAY,LTOUR,ITYPE)                                4510
C                                                                       4511
C SUBROUTINE TO DETERMINE THE EFFECT ON THE OBJECTIVE FUNCTION OF       4512
C ADDING A CAR TO A TOUR OR TAKING A CAR AWAY FROM AN OVERLAY TOUR      4513
C AND ADDING A CAR TO EACH OF THE OVERLAID TOURS.                       4514
C                                                                       4515
      COMMON/PNTRS/IOVRLY,IOVTR(2),                                     4516
     1NPCTDT,NPCTRD,LPCTDT,LNMLST(4),NNAMES(4),NDAYDT,LDAYNM,           4517
     2LDYRFL,NDAYRD,LDYWFL,NTRDT,LTRTB(2),LTRST,LTREND,LTRRFL,LTRNM,    4518
     3NTRRD,LTRWFL,NBLDT,LBLKTB(2),LBLRFL,NBLRD,LBLWFL,NDIVDT,NDIVRD,   4519
     4LDIVNM,LDIVFL                                                     4520
C                                                                       4521
      COMMON/OFFSET/NMPOFF,DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,   4522
     1NWDPCT,CPDOFF,SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,NWDDY,           4523
     2QDTOFF,QXTOFF,CRTOFF,QOTOFF,QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,   4524
     3PVTOFF,HFTOFF,MFTOFF,LFTOFF,NPRIO,NWDTR,BLDOFF,QOBOFF,QNBOFF,     4525
     4EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,NWDBL            4526
C                                                                       4527
      INTEGER DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,CPDOFF,         4528
     1SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,QDTOFF,QXTOFF,CRTOFF,QOTOFF,   4529
     2QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,PVTOFF,HFTOFF,BLDOFF,          4530
     3EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,QOBOFF,QNBOFF    4531
C                                                                       4532
C                                                                       4533
      COMMON/STORE/TOP,BOT,RDBOT,MAXBOT,NWORDS,CDAT(6000),C2DAT(6000)   4534
      INTEGER TOP,BOT,RDBOT                                             4535
      DIMENSION ICDAT(6000),IC2DAT(6000)                                4536
      EQUIVALENCE(ICDAT,CDAT),(IC2DAT,C2DAT)                            4537
```



```
C                                                                       4538
C                                                                       4539
      ISTART=ICDAT(LTRST+ITYPE-1)                                       4540
      IEND=ICDAT(LTREND+ITYPE-1)                                        4541
      TOURLN=IEND-ISTART+1                                              4542
      CDAT(LTOUR+QDTOFF)=(CDAT(LTOUR+QOTOFF)-CDAT(LTOUR+QNTOFF))/TOURLN 4543
      IF(ICDAT(LTOUR+TYTOFF) .NE. 5) RETURN                             4544
      IF(CDAT(LTOUR+ACTOFF) .GE. 1.) GO TO 10                           4545
      CDAT(LTOUR+QXTOFF)=-1.                                            4546
      RETURN                                                            4547
10    TOTLEN=0.                                                         4548
      QOLD=0.                                                           4549
      QNEW=0.                                                           4550
      DO 20 I=1,2                                                       4551
      ITRRD=IOVTR(I)                                                    4552
      ITP=ICDAT(LTRWFL+ITRRD-1)                                         4553
      IBDT=ICDAT(LTRTB(I)+ITP-1)                                        4554
      IBRD=ICDAT(LBLRFL+IBDT-1)                                         4555
      LBLK=LDAY+BLDOFF+(IBRD-1)*NWDBL                                   4556
      QOLD=QOLD+CDAT(LBLK+QOBOFF)                                       4557
      QNEW=QNEW+CDAT(LBLK+QNBOFF)                                       4558
      ISTART=ICDAT(LTRST+ITP-1)                                         4559
      IEND=ICDAT(LTREND+ITP-1)                                          4560
      TOTLEN=TOTLEN+(IEND-ISTART+1)                                     4561
20    CONTINUE                                                          4562
      CDAT(LTOUR+QXTOFF)=(QOLD-QNEW)/(TOTLEN-TOURLN)                    4563
      RETURN                                                            4564
      END                                                               4565
```

SUBROUTINE STROBJ

Subroutine STROBJ determines the contribution of one shift to the objective function value and the difference in its contribution per car-hour if an additional car were assigned to the shift. LDAY and LTOUR are pointers to the data for the day and shift. ITYPE is the tour to which the shift belongs.

The objective function contributions of the shift are determined by summing the contributions of its blocks. Subroutine STRDF is called to determine the improvement per car-hour that would be realized if one car were added to the shift. (STRDF also determines the improvement per car-hour that would be realized by removing a car from an overlay shift and adding one car to each of the shifts that it overlays.)

```
      SUBROUTINE STROBJ(LDAY,LTOUR,ITYPE)                                4566
C                                                                        4567
C  EVALUATES A WEIGHTED OBJECTIVE FUNCTION FOR ONE SHIFT.                4568
C                                                                        4569
C                                                                        4570
      COMMON/STORE/TOP,BOT,RDBOT,MAXBOT,NWORDS,CDAT(6000),C2DAT(6000)    4571
      INTEGER TOP,BOT,RDBOT                                              4572
      DIMENSION ICDAT(6000),IC2DAT(6000)                                 4573
      EQUIVALENCE(ICDAT,CDAT),(IC2DAT,C2DAT)                             4574
C                                                                        4575
C                                                                        4576
      COMMON/PNTRS/IOVRLY,IOVTR(2),                                      4577
     1NPCTDT,NPCTRD,LPCTDT,LNMLST(4),NNAMES(4),NDAYDT,LDAYNM,            4578
     2LDYRFL,NDAYRD,LDYWFL,NTRDT,LTRTB(2),LTRST,LTREND,LTRRFL,LTRNM,     4579
     3NTRRD,LTRWFL,NBLDT,LBLKTB(2),LBLRFL,NBLRD,LBLWFL,NDIVDT,NDIVRD,    4580
     4LDIVNM,LDIVFL                                                      4581
C                                                                        4582
      COMMON/OFFSET/NMPOFF,DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,    4583
     1NWDPCT,CPDOFF,SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,NWDDY,            4584
     2QDTOFF,QXTOFF,CRTOFF,QOTOFF,QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,    4585
     3PVTOFF,HFTOFF,MFTOFF,LFTOFF,NPRIO,NWDTR,BLDOFF,QOBOFF,QNBOFF,      4586
     4EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,NWDBL             4587
C                                                                        4588
      INTEGER DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,CPDOFF,          4589
     1SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,QDTOFF,QXTOFF,CRTOFF,QOTOFF,    4590
     2QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,PVTOFF,HFTOFF,BLDOFF,           4591
     3EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,QOBOFF,QNBOFF     4592
C                                                                        4593
      QOLD=0.                                                            4594
      QNEW=0.                                                            4595
      DO 10 IBLK=1,2                                                     4596
```



```
      IBDT=ICDAT(LTRTB(IBLK)+ITYPE-1)                                    4597
      IF(IBDT .LT. 1) GO TO 10                                           4598
      IBRD=ICDAT(LBLRFL+IBDT-1)                                          4599
      LBLK=LDAY+BLDOFF+(IBRD-1)*NWDBL                                    4600
      QOLD=QOLD+CDAT(LBLK+QOBOFF)                                        4601
      QNEW=QNEW+CDAT(LBLK+QNBOFF)                                        4602
10    CONTINUE                                                           4603
      CDAT(LTOUR+QOTOFF)=QOLD                                            4604
      CDAT(LTOUR+QNTOFF)=QNEW                                            4605
      CALL STRDF(LDAY,LTOUR,ITYPE)                                       4606
      RETURN                                                             4607
      END                                                                4608
```

SUBROUTINE TITLE

Subroutine TITLE is called by the routines that control DISP command table output to print column headings. Parameter ITAB specifies the table for which headings are to be printed. NAME is an array that contains eight characters in A1 format used as a heading to identify the leftmost column. NAME can identify the column entries as being tour names, precinct names, or day names (although the day name option has not been implemented).

```
      SUBROUTINE TITLE(ITAB,NAME)                                       4609
C                                                                       4610
C  PRINTS COLUMN HEADINGS FOR TABLE ITAB                                4611
C                                                                       4612
      COMMON/SYSTEM/SYSIN,SYSOUT,IFILE,LIT                              4613
      INTEGER SYSIN,SYSOUT                                              4614
      COMMON/TITLES/RTITLE(60),DTITLE(60),RUNFLG,DSNFLG                 4615
      INTEGER RUNFLG,DSNFLG                                             4616
C                                                                       4617
      DIMENSION NAME(8), TABHDR(5,20)                                   4618
C                                                                       4619
7     GO TO (10,20,30,40,50),ITAB                                       4620
C                                                                       4621
C ***************************************************************     4622
C                     TABLE 1                                   *       4623
C ***************************************************************     4624
C                                                                       4625
10      WRITE(SYSOUT,11) NAME                                           4626
C                                                                       4627
11      FORMAT(                                                         4628
     1'0                                        --- DURING THE TOUR -' 4629
     1,'--AVG. '/                                                       4630
     1'              NO.   CAR     CALL    SERV    -PERCENT TIME BUSY--', 4631
     1'   CARS'/                                                        4632
     1 2X,8A1,   '  CARS  HOURS    RATE    TIME    CFS   NONCFS   TOTAL', 4633
     1'  AVAIL'/)                                                       4634
        RETURN                                                          4635
C                                                                       4636
C ***************************************************************     4637
C                     TABLE 2                                   *       4638
C ***************************************************************     4639
C                                                                       4640
20      WRITE(SYSOUT,21) NAME                                           4641
21      FORMAT(                                                         4642
     1'0               NO.      CAR     OFFICER    UNCOMM   % TIME    ', 4643
     1'      '/                                                         4644
     1 2X,8A1,   '     CARS     HOURS    HOURS    HRS/CAR   UNCOMM    ', 4645
     1'      '/)                                                        4646
        RETURN                                                          4647
```



```
C                                                                       4648
C ***************************************************************     4649
C                     TABLE 3                                   *       4650
C ***************************************************************     4651
C                                                                       4652
30      WRITE(SYSOUT,31) NAME                                           4653
31      FORMAT(                                                         4654
     1'0              NO.   CAR   PRTY  2 DELAY   PRTY  3 DELAY   -AVERA', 4655
     1'GE   DELAY-'/                                                    4656
     1 2X,8A1,   '  CARS  HOURS   QUEUE   +TRVL   QUEUE   +TRVL   QUEUE', 4657
     1'     +TRVL'/)                                                    4658
        RETURN                                                          4659
C                                                                       4660
C ***************************************************************     4661
C                     TABLE 4                                   *       4662
C ***************************************************************     4663
C                                                                       4664
40      WRITE(SYSOUT,41) NAME                                           4665
41      FORMAT(                                                         4666
     1'0              NO.     CAR     ---- PERCENT  OF CALLS DELAYED ---', 4667
     1'-    PATROL'/                                                    4668
     1 2X,8A1,   '  CARS    HOURS   PRTY1     PRTY2     PRTY3     TOTA', 4669
     1'L    INTERVAL'/)                                                 4670
        RETURN                                                          4671
C                                                                       4672
C ***************************************************************     4673
C                     TABLE 5                                   *       4674
C ***************************************************************     4675
C                                                                       4676
50      WRITE(SYSOUT,51) NAME                                           4677
51      FORMAT(                                                         4678
     1'0              NO.   CAR    ------ AVG CARS/CFS ------   PRTY  1 ', 4679
     1' DELAY'/                                                         4680
     1 2X,8A1,   '  CARS  HOURS   PRTY1  PRTY2  PRTY3  TOTAL    QUEUE  ', 4681
     1' +TRVL'/)                                                        4682
C                                                                       4683
        RETURN                                                          4684
        END                                                             4685
```

SUBROUTINE TOTAL

Subroutine TOTAL is called from the routines that control DISP command output. Its function is to add weighted sums and weights for a specified level of output measures to the accumulators for the next higher level. Averages are computed for the specified level and printed by means of a call to PRTBL.

Parameter ITAB specifies the table of output measures being displayed. LEV specifies the level of measures to be printed. N gives the number of observations at level LEV+1 that are reflected in the level LEV sums (if N is less than 2, no level LEV statistics are printed). IADD indicates whether the accumulation of level LEV sums into LEV-1 is to take place (this depends on overlay considerations).

```
      SUBROUTINE TOTAL(ITAB,LEV,N,IADD)                                  4686
C                                                                        4687
C  ACCUMULATES SUMS FOR WEIGHTED AVERAGES IN TABLES                      4688
C                                                                        4689
      COMMON/STATS/T(4,8),S(4,8),PORDER(3),RORDER(3),CIND(8)             4690
      INTEGER PORDER,RORDER                                              4691
C                                                                        4692
      COMMON/SYSTEM/SYSIN,SYSOUT,IFILE,LIT                               4693
      INTEGER SYSIN,SYSOUT                                               4694
C                                                                        4695
      DIMENSION AV(8),ITABSZ(5)                                          4696
      DATA AV(1)/1HA/,AV(2)/1HV/,AV(3)/1HE/,AV(4)/1HR/,AV(5)/1HA/,       4697
     1 AV(6)/1HG/,AV(7)/1HE/,AV(8)/1H /,FLAG/1H /,ITABSZ(1)/8/,          4698
     1 ITABSZ(2)/5/,ITABSZ(3)/8/,ITABSZ(4)/7/,ITABSZ(5)/8/               4699
C                                                                        4700
      IF(N .LT. 1) RETURN                                                4701
      IF(LEV .LT. 2) GO TO 15                                            4702
      M=ITABSZ(ITAB)                                                     4703
      IF(ITAB .NE. 2 .AND. IADD .EQ. 0) M=2                              4704
      LEVM1=LEV-1                                                        4705
      DO 10 I=1,M                                                        4706
      T(LEVM1,I)=T(LEVM1,I)+T(LEV,I)                                     4707
10    S(LEVM1,I)=S(LEVM1,I)+S(LEV,I)                                     4708
15    M=ITABSZ(ITAB)                                                     4709
      DO 17 I=1,M                                                        4710
      IF(S(LEV,I) .GT. 0.) GO TO 16                                      4711
      T(LEV,I)=0.                                                        4712
      GO TO 17                                                           4713
16    T(LEV,I)=T(LEV,I)/S(LEV,I)                                         4714
17    CIND(I)=FLAG                                                       4715
      IF(LEV .EQ. 4 .OR. N .LT. 2) RETURN                                4716
```



```
      WRITE(SYSOUT,1)                                                    4717
1     FORMAT(2H  )                                                       4718
      CALL PRTBL(ITAB,LEV,FLAG,AV)                                       4719
C                                                                        4720
30    X=T(LEV,1)*S(LEV,1)                                                4721
      Y=T(LEV,2)*S(LEV,2)                                                4722
      Z=T(LEV,3)*S(LEV,2)                                                4723
C                                                                        4724
      GO TO (100,200,300,400,500),ITAB                                   4725
C                                                                        4726
100   WRITE(SYSOUT,101) X, Y                                             4727
101   FORMAT('  TOTAL',3X,F6.1,1X,F6.1)                                  4728
      RETURN                                                             4729
200   WRITE(SYSOUT,201) X, Y, Z                                          4730
201   FORMAT('  TOTAL',6X,F6.1,4X,F6.1,4X,F6.1)                          4731
      RETURN                                                             4732
300   WRITE(SYSOUT,301) X, Y                                             4733
301   FORMAT('  TOTAL',3X,F6.1,1X,F6.1)                                  4734
      RETURN                                                             4735
400   WRITE(SYSOUT,401) X, Y                                             4736
401   FORMAT('  TOTAL',4X,F6.1,3X,F6.1)                                  4737
      RETURN                                                             4738
500   WRITE(SYSOUT,501) X, Y                                             4739
501   FORMAT('  TOTAL',3X,F6.1,1X,F6.1)                                  4740
      RETURN                                                             4741
      END                                                                4742
```

FUNCTION TRIDSP

Function TRIDSP is the computer implementation of Green's multiple car dispatch/priority queuing model.[2] It calculates a wide variety of queuing measures and returns selected ones to the calling routine both as the value of the function and through the variables PI and TRAVT. The value of the function depends upon the objective function being evaluated (NOBJ) and the priority class (IPRIO).

LAMDA is the call rate in one hour, ST the service time, and EF the number of effective cars for which the function is to be evaluated. PHP1 and PHP2 are the proportion of calls of priority 1 and priority 2. C1, C2, and C3 are vectors that describe the dispatch policy for calls of priority 1, 2, and 3 respectively. If the number of effective patrol cars (EF) is smaller than the number to be dispatched to each call, the workload is rearranged to be handled by EF cars.

```
      FUNCTION TRIDSP(NOBJ,IPRIO,LAMDA,ST,PHP1,PHP2,EF,C1,C2,C3,PI)      4743
      DIMENSION    Q(30),QBAR(31),QBARSM(31)                             4744
      DIMENSION    C(30),CSUM(31),RVEC(31),T(31,31),T1(31,31)            4745
      DIMENSION    CC1(3),CC1SUM(31),CC2(3),CC2SUM(31)                   4746
      DIMENSION    CC3(3),CC3SUM(31),DSUM(30,30)                         4747
      DIMENSION    RST(4,30)                                             4748
      DIMENSION    ROWS(31)                                              4749
      DIMENSION    PANYW(31), ENBS(31)                                   4750
         DIMENSION C1(3), C2(3), C3(3)                                   4751
      REAL    LAMDA,MU,LAMDA1,LAMDA2,LAMDA3                              4752
      INTEGER S,SP1,SM1                                                  4753
      COMMON/TRAVEL/SQRTA,STRDNF,RV,TRAVT,AVGTT                          4754
C                                                                        4755
         DO 2 I=1,30                                                     4756
         Q(I)=0.0                                                        4757
         C(I)=0.0                                                        4758
         DO 2 J=1,30                                                     4759
         DSUM(I,J)=0.0                                                   4760
2        CONTINUE                                                        4761
         DO 3 I=1,31                                                     4762
         QBAR(I)=0.0                                                     4763
         QBARSM(I)=0.0                                                   4764
         CSUM(I)=0.0                                                     4765
         RVEC(I)=0.0                                                     4766
         CC1SUM(I)=0.0                                                   4767
         CC2SUM(I)=0.0                                                   4768
         CC3SUM(I)=0.0                                                   4769
         ROWS(I)=0.0                                                     4770
```



```
         PANYW(I)=0.0
         ENBS(I)=0.0                                                     4771
         DO 3 J=1,31                                                     4772
         T(I,J)=0.0                                                      4773
         T1(I,J)=0.0                                                     4774
3        CONTINUE                                                        4775
         DO 4 I=1,4                                                      4776
         DO 4 J=1,30                                                     4777
         RST(I,J)=0.0                                                    4778
4        CONTINUE                                                        4779
C                                                                        4780
         EBSY=0.0                                                        4781
C                                                                        4782
         DO 1 I=1,3                                                      4783
         CC1(I)=C1(I)                                                    4784
         CC2(I)=C2(I)                                                    4785
         CC3(I)=C3(I)                                                    4786
1        CONTINUE                                                        4787
C                                                                        4788
      TRIDSP=0.0                                                         4789
      PI=0.0                                                             4790
      WAIT=0.0                                                           4791
       TST=ST                                                            4792
C                                                                        4793
      MAXDIS=3                                                           4794
      MINS=EF                                                            4795
      MAXS=EF+1                                                          4796
      PHP3=1.-PHP1-PHP2                                                  4797
      IF (PHP1 .LT. 1E-4 .AND. IPRIO .EQ. 1) GO TO  4100                 4798
      IF (PHP2 .LT. 1E-4 .AND. IPRIO .EQ. 2) GO TO  4100                 4799
      IF (PHP3 .LT. 1E-4 .AND. IPRIO .EQ. 3) GO TO  4100                 4800
      IF (MINS .GE. 3) GO TO 10                                          4801
      TST= TST * (PHP1*(CC1(1)+2.*CC1(2)+3.*CC1(3)) +                    4802
     1           PHP2*(CC2(1)+2.*CC2(2)+3.*CC2(3)) +                     4803
     2           PHP3*(CC3(1)+2.*CC3(2)+3.*CC3(3)))                      4804
C                                                                        4805
      IF (MINS .EQ. 2) GO TO 710                                         4806
C                                                                        4807
C           MINS EQ 1                                                    4808
C                                                                        4809
      MAXDIS=1                                                           4810
C                                                                        4811
      CC1(1)=1.                                                          4812
      CC1(2)=0.                                                          4813
      CC1(3)=0.                                                          4814
C                                                                        4815
      CC2(1)=1.                                                          4816
      CC2(2)=0.                                                          4817
      CC2(3)=0.                                                          4818
C                                                                        4819
      CC3(1)=1.                                                          4820
      CC3(2)=0.                                                          4821
                                                                         4822
```

CONTINUED

2 OF 3

```
      CC3(3)=0.
C
      GO TO 10
C             MINS EQ 2
C
  710 MAXDIS=2
C
      TST= TST / (PHP1*(CC1(1)+2.*CC1(2)+2.*CC1(3)) +
     1           PHP2*(CC2(1)+2.*CC2(2)+2.*CC2(3)) +
     2           PHP3*(CC3(1)+2.*CC3(2)+2.*CC3(3)))
      CC1(2)=CC1(2)+CC1(3)
      CC1(3)=0.
C
      CC2(2)=CC2(2)+CC2(3)
      CC2(3)=0.
C
      CC3(2)=CC3(2)+CC2(3)
      CC3(3)=0.
C
C
   10 CONTINUE
      MU= 1./TST
         IF ( LAMDA/(EF*MU) .LT. 1E-4    ) LAMDA=0.0
      MAXSP1=MAXS+1
      DO 20 IP1=1,MAXSP1
      DO 20 JP1=1,MAXSP1
   20 T(IP1,JP1)=0.
      DO 40 J=1,MAXDIS
      C(J)=PHP1*CC1(J)+PHP2*CC2(J)+PHP3*CC3(J)
   40 T(1,J+1)=C(J)
         RVEC(1)=0.0
         IF ( LAMDA .NE. 0.0 )
     1RVEC(1)=1./LAMDA
      DO 50 I=1,MAXS
         DO 45 K=1,I
         DSUM(I,K)=0.
         JK=I-K+1
         DO 44 J=JK,I
   44       DSUM(I,K)=DSUM(I,K)+1./J
   45    DSUM(I,K)=DSUM(I,K)/MU
      RVEC(I+1)=1./(LAMDA+I*MU)
      T(I+1,I)=I*MU*RVEC(I+1)
      FAC=LAMDA*RVEC(I+1)
      IF (I.EQ.MAXS) GOTO 50
      IP1=I+1
         DO 60 J=IP1,MAXS
         IF(C(J-I) .LT. 1E-4 .OR. FAC .LT. 1E-4) GO TO 60
         T(I+1,J+1)=C(J-I)*FAC
   60    CONTINUE
   50 CONTINUE
      S=MINS
```



```
   65 CONTINUE
      SP1=S+1
      DO 75 IP1=1,SP1
      DO 70 JP1=1,SP1
   70 T1(IP1,JP1)=-T(IP1,JP1)
   75 T1(IP1,IP1)=1.

      DO 820 IP1=1,SP1
  820 ROWS(IP1)=1.
      DO 850 I=1,S
         PIVM = 0.0
         IF ( T1(I,I) .NE. 0.0 )
     1PIVM=-T1(I+1,I)/T1(I,I)
         DO 830 J=I,S
  830    T1(I+1,J+1)=T1(I+1,J+1)+PIVM*T1(I,J+1)
         DO 840 JP1=1,I
  840    ROWS(JP1)=PIVM*ROWS(JP1)
  850 CONTINUE
      EQBAR=0.
      DO 860 JP1=1,SP1
         IF ( T1(S+1,S+1) .NE. 0.0 )
     1ROWS(JP1)=ROWS(JP1)/T1(S+1,S+1)
  860 EQBAR=EQBAR+ROWS(JP1)*RVEC(JP1)
      ELBAR=EQBAR*LAMDA
         PD = 0.0
         IF ( ELBAR .NE. 0.0 )
     1PD=1./ELBAR
      DO 100 IP1=1,SP1
         IF ( EQBAR .NE. 0.0 )
     1QBAR(IP1)=RVEC(IP1)*ROWS(IP1)/EQBAR
  100    CONTINUE

      II=S-MAXDIS
      CSUM(II+1)=0.
      CC1SUM(II+1)=0.
      CC2SUM(II+1)=0.
      CC3SUM(II+1)=0.
      PD1=0.
      PD2=0.
      PD3=0.
      ITEMP=MAXDIS
      I0=S-MAXDIS+1
      DO 105 I=I0,S
      CSUM(I+1)=CSUM(I)+C(ITEMP)
      CC1SUM(I+1)=CC1SUM(I)+CC1(ITEMP)
      CC2SUM(I+1)=CC2SUM(I)+CC2(ITEMP)
      CC3SUM(I+1)=CC3SUM(I)+CC3(ITEMP)
      PD1=PD1+QBAR(I+1)*CC1SUM(I+1)
      PD2=PD2+QBAR(I+1)*CC2SUM(I+1)
      PD3=PD3+QBAR(I+1)*CC3SUM(I+1)
  105 ITEMP=ITEMP-1
```

```
      ED=0.                                                              4927
      ED1=0.                                                             4928
      ED2=0.                                                             4929
      ED3=0.                                                             4930
      QBARSM(1)=0.                                                       4931
      DO 110 I=1,S                                                       4932
      QBARSM(I+1)=QBARSM(I)+QBAR(I+1)                                    4933
         KTOP=MAXDIS+I-S                                                 4934
         IF (KTOP .LT. 1) GO TO 110                                      4935
         DO 120 K=1,KTOP                                                 4936
         ED1=ED1+DSUM(I,K)*QBAR(I+1)*CC1(S-I+K)/PD1                      4937
         ED2=ED2+DSUM(I,K)*QBAR(I+1)*CC2(S-I+K)/PD2                      4938
         ED3=ED3+DSUM(I,K)*QBAR(I+1)*CC3(S-I+K)/PD3                      4939
         IF ( PD .NE. 0.0 )                                              4940
     1   ED=ED+DSUM(I,K)*QBAR(I+1)*C(S-I+K)/PD                           4941
120      CONTINUE                                                        4942
 110 CONTINUE                                                            4943
                                                                         4944
      EB=0.                                                              4945
      EB1=0.                                                             4946
      EB2=0.                                                             4947
      EB3=0.                                                             4948
      DO 140 K=1,MAXDIS                                                  4949
      EB1=EB1+DSUM(S,K)*CC1(K)                                           4950
      EB2=EB2+DSUM(S,K)*CC2(K)                                           4951
      EB3=EB3+DSUM(S,K)*CC3(K)                                           4952
  140 EB=EB+DSUM(S,K)*C(K)                                               4953
      IF (LAMDA*EB.GE.1.) GOTO 4000                                      4954
C                                                                        4955
      EQ=ED/(1.-LAMDA*EB)                                                4956
      PQ=EQ/(EQBAR+EQ)                                                   4957
      EL=LAMDA*EQ                                                        4958
      ENBS(S)=0.                                                         4959
      IF (S.EQ.1) GOTO 146                                               4960
      SM1=S-1                                                            4961
      DO 145 I=1,SM1                                                     4962
      PISB=QBAR(I+1)*(1.-PQ)                                             4963
  145 ENBS(S)=ENBS(S)+I*PISB                                             4964
  146 ENBS(S)=ENBS(S) + S*(QBAR(S+1)*(1.-PQ) +PQ)                        4965
C                                                                        4966
C     PROBABILITY OF WAITING BEFORE DISPATCH OF ALL NEEDED CARS          4967
                                                                         4968
      PDEL=PQ + (1.-PQ)*PD                                               4969
C                                                                        4970
C                                                                        4971
C     GREEN "PROBABILITY OF INITIAL DELAY" -- PINDEL                     4972
C      IS HERE CALLED "PROBABILITY OF ANY WAIT" -- PANYW                 4973
      PANYW(S)= PQ + (1. -PQ)*QBAR(S+1)                                  4974
C                                                                        4975
 141  IF (PDEL.GT.1E-4) GOTO 165                                         4976
      IF (S.EQ.MINS) GOTO 4100                                           4977
      GOTO 4200                                                          4978
```



```
  165 CONTINUE                                                           4979
                                                                         4980
      ERB1=0.                                                            4981
      ERB2=0.                                                            4982
      ERB3=0.                                                            4983
      QSUM=QBARSM(S+1)-QBARSM(II+1)                                      4984
      IO=S-MAXDIS+1                                                      4985
      DO 550 I=IO,S                                                      4986
         XXFF = 0.0                                                      4987
         IF ( PD .NE. 0.0 )                                              4988
     1   XXFF = QSUM*CSUM(I+1)/PD                                        4989
      Q(I)=(EL*CSUM(I+1) + XXFF) / (I*MU*EQ)                             4990
      TERM1=0.                                                           4991
      TERM2=0.                                                           4992
      TERM3=0.                                                           4993
         JTOP=MAXDIS+I-S                                                 4994
         DO 540 J=1,JTOP                                                 4995
         TERM1=TERM1+DSUM(I,J)*Q(I)*CC1(S-I+J)                           4996
         TERM2=TERM2+DSUM(I,J)*Q(I)*CC2(S-I+J)                           4997
         TERM3=TERM3+DSUM(I,J)*Q(I)*CC3(S-I+J)                           4998
  540 CONTINUE                                                           4999
      IF (CC1SUM(I+1).GT.0.) ERB1=ERB1+TERM1/CC1SUM(I+1)                 5000
      IF (CC2SUM(I+1).GT.0.) ERB2=ERB2+TERM2/CC2SUM(I+1)                 5001
      IF (CC3SUM(I+1).GT.0.) ERB3=ERB3+TERM3/CC3SUM(I+1)                 5002
  550 QSUM=QSUM-QBAR(I+1)                                                5003
                                                                         5004
C-------- FINAL CALCULATIONS ----------                                  5005
      LAMDA1=PHP1*LAMDA                                                  5006
      LAMDA2=PHP2*LAMDA                                                  5007
      LAMDA3=PHP3*LAMDA                                                  5008
      PQ1=LAMDA1*( PQ*EB1 + (1.-PQ)*PD1*ED1 )                            5009
      PQ2=LAMDA2*( PQ*EB2 + (1.-PQ)*PD2*ED2 )                            5010
      PQ3=LAMDA3*( PQ*EB3 + (1.-PQ)*PD3*ED3 )                            5011
      AD=1.-Q(S)                                                         5012
      ER1=PD1*ED1*(1.-PQ)+ERB1*PQ1+(EB1+ERB2*AD)*PQ2+(EB1+ERB3*AD)*PQ3   5013
      EW1=ER1/(1. - LAMDA1*EB1)                                          5014
      SMU=1./(S*MU)                                                      5015
      DIFF1=(EB1-SMU)*PQ + (ED1*PD1-QBAR(S+1)*SMU)*(1.-PQ)               5016
      DIFF2=(ED2*PD2-SMU*QBAR(S+1))*(1.-PQ) + (EB2-SMU)*PQ               5017
      ER2=ED2*PD2*(1.-PQ)+(ERB1+EB2-EB1)*PQ1+ERB2*PQ2+(EB2+ERB3*AD)*PQ3  5018
     1-LAMDA1*EB1*DIFF2                                                  5019
      EW2=(LAMDA1*EW1*EB1 + ER2)/(1.-LAMDA1*EB1-LAMDA2*EB2)              5020
      DIFF3=(ED3*PD3-SMU*QBAR(S+1))*(1.-PQ) + (EB3-SMU)*PQ               5021
      ER3=PD3*ED3*(1-PQ)+ERB3*PQ3+(ERB1-EB1+EB3)*PQ1+(ERB2-EB2+EB3)*PQ2  5022
     1-LAMDA1*EB1*DIFF3-LAMDA2*EB2*DIFF3                                 5023
      EW3=(LAMDA1*EW1*EB1+LAMDA2*EW2*EB2+ER3)/(1.-LAMDA1*EB1-LAMDA2*EB2  5024
     1    -LAMDA3*EB3)                                                   5025
      ERES1=EW1-DIFF1                                                    5026
      ERES2=EW2-DIFF2                                                    5027
      ERES3=EW3-DIFF3                                                    5028
      ERES=PHP1*ERES1 + PHP2*ERES2 + PHP3*ERES3                          5029
                                                                         5030
```

```
      RST(1,S)=60.0*ERES                                                    5031
      RST(2,S)=60.0*ERES1                                                   5032
      RST(3,S)=60.0*ERES2                                                   5033
      RST(4,S)=60.0*ERES3                                                   5034
 1000 CONTINUE                                                              5035
      S=S+1                                                                 5036
      IF (S.LE.MAXS) GOTO 65                                                5037
 1001 PI=PANYW(MAXS) + (MAXS - EF) * (PANYW(MINS) - PANYW(MAXS))            5038
      IF (NOBJ .GT. 1) GOTO 2000                                            5039
      TRIDSP=PI                                                             5040
 1002 RETURN                                                                5041
 2000 WAIT=RST(IPRIO+1,MAXS)                                                5042
     1 +(MAXS-EF)*(RST(IPRIO+1,MINS)-RST(IPRIO+1,MAXS))                     5043
      EBSY=ENBS(MAXS) + (MAXS -EF) * (ENBS(MINS) - ENBS(MAXS))              5044
      IF (NOBJ .GT. 2) GOTO 5000                                            5045
      TRIDSP=WAIT                                                           5046
      IF(NOBJ .EQ. 1) RETURN                                                5047
 2002 GO TO 5000                                                            5048
 4000 IF (NOBJ .EQ. 1)TRIDSP=1.0                                            5049
      PI=1.0                                                                5050
      IF (NOBJ .GT. 1)TRIDSP=9.E65                                          5051
      IF(NOBJ .EQ. 1) RETURN                                                5052
 4002 GO TO 5005                                                            5053
 4100 WAIT=0.                                                               5054
      PI=0.                                                                 5055
      EBSY=ENBS(MINS)                                                       5056
      IF (NOBJ .GT. 2) GO TO 5000                                           5057
      TRIDSP=WAIT                                                           5058
      IF(NOBJ .EQ. 1) RETURN                                                5059
 4102 GO TO 5000                                                            5060
 4200 WAIT   =(MAXS-EF)*RST(IPRIO+1,MINS)                                   5061
      PI=(MAXS-EF)*PANYW(MINS)                                              5062
      EBSY=(MAXS-EF)*ENBS(MINS)                                             5063
      IF (NOBJ .GT. 2) GO TO 5000                                           5064
      TRIDSP=WAIT                                                           5065
      IF(NOBJ .EQ. 1) RETURN                                                5066
 5000 AVAVL=EF-EBSY                                                         5067
      IF(AVAVL .GE.  1.) GO TO 5010                                         5068
 5005  TD=.678*SQRTA                                                       5069
      GO TO 5040                                                            5070
 5010 IF (AVAVL .GE.. 2.) GO TO 5020                                        5071
      TD=SQRTA*(.08+.598/SQRT(AVAVL))                                       5072
      GO TO 5040                                                            5073
 5020 TD=.711*SQRTA/SQRT(AVAVL)                                             5074
 5040 TRAVT=TD*STRDNF/RV*60.0                                               5075
 5042 RETURN                                                                5076
C                                                                           5077
      END                                                                   5078
```


FUNCTION WLEFT

Function WLEFT is called by function OBJF2 to obtain the waiting time of calls in the first hour of a block, when the option of smoothing queuing behavior over time has been selected. It uses function TRIDSP to calculate the waiting time. Parameter N specifies the priority level of interest.

```
      FUNCTION WLEFT(N,CR,ST,PHP1,PHP2,EF,EFM1,C1,C2,C3,LCR,LST,            5079
     1       LPR,ISTART)                                                    5080
C                                                                           5081
C     WAITING TIME AND TRAVEL TIME FOR FIRST HOUR OF BLOCK                  5082
C                                                                           5083
      COMMON/STORE/TOP,BOT,RDBOT,MAXBOT,NWORDS,CDAT(6000),C2DAT(6000)       5084
      INTEGER TOP,BOT,RDBOT                                                 5085
      DIMENSION ICDAT(6000),IC2DAT(6000)                                    5086
      EQUIVALENCE(ICDAT,CDAT),(IC2DAT,C2DAT)                                5087
C                                                                           5088
      DIMENSION C1(3),C2(3),C3(3)                                           5089
C                                                                           5090
      COMMON/TRAVEL/SQRTA,STRDNF,SPEED,TRAVT,AVGTT                          5091
      CR=CDAT(LCR+ISTART-1)                                                 5092
      ST=CDAT(LST+ISTART-1)                                                 5093
      CRM1=CDAT(LCR+ISTART-2)                                               5094
      STM1=CDAT(LST+ISTART-2)                                               5095
      LPRIHR=LPR+ISTART                                                     5096
      PDM1=C2DAT(LPRIHR-2)                                                  5097
      CR1=CR+CRLEFT(CRM1,STM1,PDM1,EFM1,PHP1,PHP2,C1,C2,C3)                 5098
      WLEFT=TRIDSP(2,N,CR1,ST,PHP1,PHP2,EF,C1,C2,C3,PI)                     5099
      PD1=PDEL(1,LPRIHR,CR,CR1,ST,PHP1,PHP2,EF,C1,C2,C3,PI)                 5100
      C2DAT(LPR+ISTART-1) = PD1                                             5101
      RETURN                                                                5102
      END                                                                   5103
                                                                            5104
                                                                            5105
```

SUBROUTINE WRITE

Subroutine WRITE carries out the WRITE command. It writes a file on a user-specified unit number that can later be used as a DATABASE file. Data are written only for precincts, days, and tours specified in the command qualifier.

```
      SUBROUTINE WRITE                                                  5106
C                                                                       5107
C SUBROUTINE IMPLEMENTS WRITE COMMAND                                   5108
C                                                                       5109
      COMMON/KEYWDS/NKYWD,NTYPES,TYPOFF(4),KEYWD(8,30),WDTYPE(30)       5110
      INTEGER TYPOFF,WDTYPE                                             5111
      DIMENSION PCLSNM(8),DCLSNM(8),TOURNM(8)                           5112
      EQUIVALENCE (PCLSNM,KEYWD(1,4)),(DCLSNM,KEYWD(1,3)),              5113
     1(TOURNM,KEYWD(1,2))                                               5114
C                                                                       5115
C                                                                       5116
      COMMON/STORE/TOP,BOT,RDBOT,MAXBOT,NWORDS,CDAT(6000),C2DAT(6000)   5117
      INTEGER TOP,BOT,RDBOT                                             5118
      DIMENSION ICDAT(6000),IC2DAT(6000)                                5119
      EQUIVALENCE(ICDAT,CDAT),(IC2DAT,C2DAT)                            5120
C                                                                       5121
C                                                                       5122
      COMMON/OFFSET/NMPOFF,DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,   5123
     1NWDPCT,CPDOFF,SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,NWDDY,           5124
     2QDTOFF,QXTOFF,CRTOFF,QOTOFF,QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,   5125
     3PVTOFF,HFTOFF,MFTOFF,LFTOFF,NPRIO,NWDTR,BLDOFF,QOBOFF,QNBOFF,     5126
     4EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,NWDBL            5127
C                                                                       5128
      INTEGER DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,CPDOFF,         5129
     1SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,QDTOFF,QXTOFF,CRTOFF,QOTOFF,   5130
     2QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,PVTOFF,HFTOFF,BLDOFF,          5131
     3EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,QOBOFF,QNBOFF    5132
C                                                                       5133
      COMMON/PNTRS/IOVRLY,IOVTR(2),                                     5134
     1NPCTDT,NPCTRD,LPCTDT,LNMLST(4),NNAMES(4),NDAYDT,LDAYNM,           5135
     2LDYRFL,NDAYRD,LDYWFL,NTRDT,LTRTB(2),LTRST,LTREND,LTRRFL,LTRNM,    5136
     3NTRRD,LTRWFL,NBLDT,LBLKTR(2),LBLRFL,NBLRD,LBLWFL,NDIVDT,NDIVRD,   5137
     4LDIVNM,LDIVFL                                                     5138
C                                                                       5139
      COMMON/SYSTEM/SYSIN,SYSOUT,IFILE,LIT                              5140
      INTEGER SYSIN,SYSOUT                                              5141
C                                                                       5142
      COMMON/SCODES/SEND,CMD,NUMLST,NAMLST,FSPEC,DSPEC,DUM,ERR          5143
      INTEGER SEND,CMD,FSPEC,DSPEC,DUM,ERR                              5144
C                                                                       5145
      INTEGER TYPE,VAL                                                  5146
      DIMENSION VAL(2),ORDER(3)                                         5147
      DATA BLANK/1H /                                                   5148
```



```
C                                                                       5149
      COMMON/OPTION/NOERCK                                              5150
      COMMON/TITLES/RTITLE(60),DTITLE(60),RUNFLG,DSNFLG                 5151
C     INTEGER RUNFLG,DSNFLG                                             5152
      LGETT=TOP                                                         5153
      TYPE=CMD                                                          5154
      CALL SCAN(TYPE,VAL)                                               5155
C                                                                       5156
C          GET UNIT NUMBER                                              5157
C                                                                       5158
      IF(TYPE .EQ. NUMLST) GO TO 20                                     5159
10    WRITE(SYSOUT,9010)                                                5160
9010  FORMAT('0**** INVALID UNIT SPECIFICATION - REENTER.')             5161
      TOP=LGETT                                                         5162
      RETURN                                                            5163
20    NUNIT=ICDAT(VAL(2))                                               5164
      IF(NUNIT .EQ. SYSIN .OR. NUNIT .EQ. SYSOUT .OR. NUNIT .EQ. IFILE  5165
     1 .OR. NUNIT .LT. 1 .OR. NUNIT .GT. 99) GO TO 10                   5166
C                                                                       5167
C          SCAN QUALIFIER                                               5168
C                                                                       5169
      CALL SCAN(TYPE,VAL)                                               5170
      CALL GTDSPC(TYPE,VAL,ORDER)                                       5171
      IF(TYPE .NE. ERR) GO TO 30                                        5172
      TOP=LGETT                                                         5173
      RETURN                                                            5174
C                                                                       5175
C          SET WORK FLAGS                                               5176
C                                                                       5177
30    CALL SETWFL(IERR)                                                 5178
      IF(IERR .EQ. 0)GO TO 35                                           5179
      TOP=LGETT                                                         5180
      RETURN                                                            5181
C                                                                       5182
C          CHECK OVERLAY SPECIFICATION                                  5183
C                                                                       5184
35    CALL CKOVR(IERR)                                                  5185
      IF(IERR .EQ. 0) GO TO 40                                          5186
      TOP=LGETT                                                         5187
      RETURN                                                            5188
C                                                                       5189
C          DETERMINE NUMBER OF DAY, TOURS AND PRECINCTS                 5190
C          IN NEW DATA BASE                                             5191
C                                                                       5192
40    NDAY=NNAMES(1)                                                    5193
      IF(NDAY .LT. 1) NDAY=NDAYRD                                       5194
      NTOUR=NNAMES(2)                                                   5195
      IF(NTOUR .LT. 1) NTOUR =NTRRD                                     5196
      NPCT=0                                                            5197
      LPCT=0                                                            5198
50    LPCT=NXPCT(LPCT)                                                  5199
                                                                        5200
```

```
      IF(LPCT .LE. 0) GO TO 55                                          5201
      NPCT=NPCT+1                                                       5202
      GO TO 50                                                          5203
55    CONTINUE                                                          5204
      IOVT=IOVRLY                                                       5205
      IF(ICDAT(LTRRFL+NTRDT-1) .EQ. 0 .OR. (ICDAT(LTRRFL+NTRDT-1)      5206
     1  .EQ. 1 .AND. ICDAT(LTRWFL+NTRRD-1) .EQ. 0)) IOVT=0              5207
C                                                                       5208
C          WRITE CONTROL RECORD                                         5209
C                                                                       5210
      WRITE(NUNIT,1) DCLSNM,PCLSNM,TOURNM,NDIVRD,NPCT,NDAY,NBLDT,      5211
     1 NTOUR,IOVT,NOERCK,IC2DAT(1),DSNFLG                               5212
      IF(DSNFLG .EQ. 1) WRITE(NUNIT,11) DTITLE                          5213
      CALL GETTOP(80,LREC)                                              5214
      I=0                                                               5215
      K=LREC-1                                                          5216
      DO 60 IDAY=1,NDAYRD                                               5217
      ID=ICDAT(LDYWFL+IDAY-1)                                           5218
      IF(ID .LT. 1) GO TO 60                                            5219
      LNM=(ID-1)*8+LDAYNM                                               5220
      CALL MOVE(ICDAT(LNM),ICDAT(LREC+I),8)                             5221
      I=I+8                                                             5222
      IF(I .LT. 80) GO TO 60                                            5223
C                                                                       5224
C          WRITE DAY NAMES                                              5225
C                                                                       5226
      WRITE(NUNIT,2)(ICDAT(K+J),J=1,80)                                 5227
      I=0                                                               5228
60    CONTINUE                                                          5229
      IF(I .GT. 0) WRITE(NUNIT,2) (ICDAT(K+J),J=1,I)                    5230
      K=LBLKTB(2)-1                                                     5231
C                                                                       5232
C          WRITE BLOCK DESCRIPTOR RECORDS                               5233
C                                                                       5234
      WRITE(NUNIT,3) (ICDAT(K+I),I=1,NBLDT)                             5235
      DO 70 ITOUR=1,NTRRD                                               5236
      IT=ICDAT(LTRWFL+ITOUR-1)                                          5237
      IF(IT .LT. 1) GO TO 70                                            5238
      IT=IT-1                                                           5239
      LNM=IT*8+LTRNM-1                                                  5240
C                                                                       5241
C          WRITE TOUR DESCRIPTOR RECORDS                                5242
C                                                                       5243
      WRITE(NUNIT,4) (ICDAT(LNM+I),I=1,8),ICDAT(LTRTB(1)+IT),           5244
     1 ICDAT(LTRTB(2)+IT)                                               5245
70    CONTINUE                                                          5246
C                                                                       5247
      LPCT=0                                                            5248
100   LPCT=NXPCT(LPCT)                                                  5249
      IF(LPCT .NE. 0) GO TO 110                                         5250
      ENDFILE NUNIT                                                     5251
      TOP=LGETT                                                         5252
```



```
      RETURN                                                            5253
110   IDIV=ICDAT(LPCT+DVPOFF)-1                                         5254
      LPCTNM=LPCT+NMPOFF-1                                              5255
      LDVNM=LDIVNM+IDIV*8-1                                             5256
C                                                                       5257
C          WRITE PRECINCT HEADER                                        5258
C                                                                       5259
      WRITE(NUNIT,5) (ICDAT(LPCTNM+I),I=1,8),(ICDAT(LDVNM+I),I=1,8),    5260
     1 (CDAT(LPCT+I),I=ARPOFF,B2POFF)                                   5261
C                                                                       5262
      LDAY=0                                                            5263
200   LDAY=NXDAY(LPCT,LDAY)                                             5264
      IF(LDAY .LT. 1) GO TO 100                                         5265
      CPARM=CDAT(LDAY+CPDOFF)                                           5266
      SPARM=CDAT(LDAY+SPDOFF)                                           5267
C                                                                       5268
C          WRITE DAY DETAIL RECORDS                                     5269
C                                                                       5270
      WRITE(NUNIT,6) CPARM,SPARM,ICDAT(LDAY+OVDOFF)                     5271
      LCRO=LREC-1                                                       5272
      LSTO=LREC+23                                                      5273
      LCRI=LDAY+CRDOFF-1                                                5274
      LSTI=LDAY+STDOFF-1                                                5275
      SPARM=SPARM/60.                                                   5276
      DO 210 I=1,24                                                     5277
      ICDAT(LCRO+I)=100.*(.005+CDAT(LCRI+I)/CPARM)                      5278
210   ICDAT(LSTO+I)=100.*(.005+CDAT(LSTI+I)/SPARM)                      5279
      WRITE(NUNIT,7) (ICDAT(LCRO+I),I=1,48)                             5280
      DO 220 I=1,24                                                     5281
220   ICDAT(LCRO+I)=0                                                   5282
C                                                                       5283
      LTOUR=0                                                           5284
300   LTOUR=NXTOUR(LDAY,LTOUR,ITYPE)                                    5285
      IF(LTOUR .NE. 0) GO TO 320                                        5286
      IF(IOVT .NE. 0 .AND. ICDAT(LDAY+OVDOFF) .EQ. 0)                   5287
     1  WRITE(NUNIT,2) BLANK                                            5288
      DO 310 I=1,NBLDT                                                  5289
      IBLK=ICDAT(LBLRFL+I-1)                                            5290
      IF(IBLK .LE. 0) GO TO 310                                         5291
      LBLK=LDAY+BLDOFF+(IBLK-1)*NWDBL                                   5292
      CDAT(LCRO+I)=CDAT(LBLK+OCBOFF)                                    5293
310   CONTINUE                                                          5294
C                                                                       5295
C          WRITE BLOCK DETAIL RECORD                                    5296
C                                                                       5297
      WRITE(NUNIT,8) (CDAT(LCRO+I),I=1,NBLDT)                           5298
      GO TO 200                                                         5299
C                                                                       5300
C          WRITE SHIFT DETAIL RECORD                                    5301
C                                                                       5302
320   WRITE(NUNIT,9) (CDAT(LTOUR+I),I=ACTOFF,MFTOFF),                   5303
     1 C2DAT(LTOUR+QDTOFF+9),(C2DAT(LTOUR+QDTOFF+I),I=1,2),             5304
```

```
     1                    (C2DAT(LTOUR+QOTOFF+I),I=1,2),                    5305
     1                    (C2DAT(LTOUR+TYTOFF+I),I=1,2)                     5306
       GO TO 300                                                           5307
1      FORMAT(2(8A1,2X),8A1,1X,I2,1X,I3,1X,I3,1X,2(I2,1X),4(I1,1X))        5308
C                                                                          5309
2      FORMAT(80A1)                                                        5310
3      FORMAT(24(I2,1X))                                                   5311
4      FORMAT(8A1,1X,I2,1X,I2)                                             5312
5      FORMAT(8A1,1X,8A1,2X,F5.2,1X,F5.1,2(1X,F5.3))                       5313
6      FORMAT(2(F5.2,1X),I1)                                               5314
7      FORMAT(24I3)                                                        5315
8      FORMAT(24F3.1)                                                      5316
C                                                                          5317
9      FORMAT(3(F5.1,1X),2(F5.3,1X),F5.1,1X,6(F5.3,1X))                    5318
11     FORMAT(60A1)                                                        5319
C                                                                          5320
C ***********************************************************************  5321
C                                                                          5322
       END                                                                 5323
```


SUBROUTINE ZERO

Subroutine ZERO is called by the subroutines that control table output to clear level N accumulators. ZERO initializes array T, which is used to accumulate weighted sums for output measures, and array S, which is used to accumulate weights.

```
       SUBROUTINE ZERO(N)                                                  5324
C                                                                          5325
C INITIALIZE ACCUMULATORS FOR LEVEL 'N' TABLE OUTPUT                       5326
C                                                                          5327
       COMMON/STATS/T(4,8),S(4,8),PORDER(3),RORDER(3),CIND(8)              5328
       INTEGER PORDER,RORDER                                               5329
C                                                                          5330
       DO 10 I=1,8                                                         5331
       S(N,I)=0.                                                           5332
10     T(N,I)=0.                                                           5333
       RETURN                                                              5334
       END                                                                 5335
```

BLOCK DATA

BLOCK DATA establishes the initial numerical values of variables in COMMON blocks used by the program's subroutines and functions.

The COMMON blocks are as follows:

COMMON/PNTRS/	Pointers. See Sec. IV.
COMMON/OFFSET/	Offsets. See Sec. IV.
COMMON/STORE/	Parameters related to run-time storage requirements.
COMMON/SYSTEM/	Input and output unit numbers.
COMMON/KEYWDS/	Keywords and word types.
COMMON/LCODES/	Lexical types returned by GETTKN.
COMMON/SCODES/	Syntactic types returned by SCAN.
COMMON/STATS/	Statistics and output orders.

```
      BLOCK DATA                                                          5336
C                                                                         5337
      COMMON/PNTRS/IOVRLY,IOVTR(2),                                       5338
     1NPCTDT,NPCTRD,LPCTDT,LNMLST(4),NNAMES(4),NDAYDT,LDAYNM,             5339
     2LDYRFL,NDAYRD,LDYWFL,NTRDT,LTRTB(2),LTRST,LTREND,LTRRFL,LTRNM,      5340
     3NTRRD,LTRWFL,NBLDT,LBLKTB(2),LBLRFL,NBLRD,LBLWFL,NDIVDT,NDIVRD,     5341
     4LDIVNM,LDIVFL                                                       5342
      DATA NDAYRD/0/,NTRRD/0/                                             5343
C                                                                         5344
      COMMON/OFFSET/NMPOFF,DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,     5345
     1NWDPCT,CPDOFF,SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,NWDDY,             5346
     2QDTOFF,QXTOFF,CRTOFF,QOTOFF,QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,     5347
     3PVTOFF,HFTOFF,MFTOFF,LFTOFF,NPRIO,NWDTR,BLDOFF,QOBOFF,QNBOFF,       5348
     4EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,NWDBL              5349
C                                                                         5350
      INTEGER DVPOFF,ARPOFF,SMPOFF,B1POFF,B2POFF,DYPOFF,CPDOFF,           5351
     1SPDOFF,OVDOFF,CRDOFF,STDOFF,TRDOFF,QDTOFF,QXTOFF,CRTOFF,QOTOFF,     5352
     2QNTOFF,CTTOFF,TYTOFF,ACTOFF,RVTOFF,PVTOFF,HFTOFF,BLDOFF,            5353
     3EFBOFF,ACBOFF,AWBOFF,CRBOFF,RMBOFF,OCBOFF,CTBOFF,QOBOFF,QNBOFF      5354
C                                                                         5355
      DATA NMPOFF/0/,DVPOFF/8/,ARPOFF/9/,SMPOFF/10/,B1POFF/11/,           5356
     1B2POFF/12/,DYPOFF/13/,CPDOFF/0/,SPDOFF/1/,OVDOFF/2/,CRDOFF/3/,     5357
     2STDOFF/27/,TRDOFF/51/,QDTOFF/0/,QXTOFF/1/,CRTOFF/2/,               5358
     3QOTOFF/3/,QNTOFF/4/,CTTOFF/5/,TYTOFF/6/,ACTOFF/7/,RVTOFF/8/,       5359
     4PVTOFF/9/,HFTOFF/10/,MFTOFF/11/,LFTOFF/12/,NPRIO/3/,NWDTR/13/,     5360
     4EFBOFF/0/,ACBOFF/1/,AWBOFF/2/,CRBOFF/3/,RMBOFF/4/,OCBOFF/5/,       5361
     5CTBOFF/6/,QOBOFF/7/,QNBOFF/8/,NWDBL/9/                              5362
C                                                                         5363
C                                                                         5364
      COMMON/STORE/TOP,BOT,RDBOT,MAXBOT,NWORDS,CDAT(6000),C2DAT(6000)     5365
      INTEGER TOP,BOT,RDBOT                                               5366
      DIMENSION ICDAT(6000),IC2DAT(6000)                                  5367
      EQUIVALENCE(ICDAT,CDAT),(IC2DAT,C2DAT)                              5368
C                                                                         5369
```



```
C     DATA BOT/1/,NWORDS/6000/,MAXBOT/0/                                  5370
      COMMON/SYSTEM/SYSIN,SYSOUT,IFILE,LIT                                5371
      INTEGER SYSIN,SYSOUT                                                5372
      DATA SYSIN/10/,SYSOUT/11/,IFILE/19/,LIT/20/                         5373
C                                                                         5374
      COMMON/KEYWDS/NKYWD,NTYPES,TYPOFF(4),KEYWD(8,30),WDTYPE(30)         5375
      INTEGER TYPOFF,WDTYPE                                               5376
      DIMENSION PCLSNM(8),DCLSNM(8),TOURNM(8)                             5377
      EQUIVALENCE (PCLSNM,KEYWD(1,4)),(DCLSNM,KEYWD(1,3)),                5378
     1(TOURNM,KEYWD(1,2))                                                 5379
      DATA NTYPES/4/,NKYWD/28/,TYPOFF(1)/4/,TYPOFF(2)/5/,TYPOFF(3)/10/,   5380
     C TYPOFF(4)/9/                                                       5381
C                                                                         5382
      COMMON/LCODES/LEND,WORD,NUM,LP,RP                                   5383
      INTEGER WORD,RP                                                     5384
      DATA LEND/1/,WORD/2/,NUM/3/,LP/4/,RP/5/                             5385
C                                                                         5386
      COMMON/SCODES/SEND,CMD,NUMLST,NAMLST,FSPEC,DSPEC,DUM,ERR            5387
      INTEGER SEND,CMD,FSPEC,DSPEC,DUM,ERR                                5388
      DATA SEND/5/,CMD/3/,NUMLST/6/,NAMLST/7/,FSPEC/2/,DSPEC/1/,          5389
     1DUM/4/,ERR/8/                                                       5390
C                                                                         5391
      COMMON/STATS/T(4,8),S(4,8),PORDER(3),RORDER(3),CIND(8)              5392
      INTEGER PORDER,RORDER                                               5393
      DATA PORDER(1)/3/,PORDER(2)/2/,PORDER(3)/1/                         5394
      COMMON/OPTION/NOERCK                                                5395
      DATA NOERCK/0/                                                      5396
      COMMON/BUFFER/CARD(81),COL                                          5397
      DATA CARD(81)/1H /                                                  5398
      COMMON/TITLES/RTITLE(60),DTITLE(60),RUNFLG,DSNFLG                   5399
      INTEGER RUNFLG,DSNFLG                                               5400
      DATA RUNFLG/0/,DSNFLG/0/                                            5401
      END                                                                 5402
                                                                          5403
```

Appendix A

DEMONSTRATION DATABASE

```
Column rule
         1111111111222222222233333333334444444444555555555566666666667777777
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345
```

```
DIVISION  PRECINCT  TOUR     02 005 007 05 04 1      1
DEMONSTRATION DATABASE AS LISTED IN R-3087/3
SUN-MON MON-TUE TUE-WED WED-THU THU-FRI FRI-SAT SAT-SUN
08 11 16 19 24
MIDDAY   01
PM       02 03
AM       04 05
FOURTH   03 04
EAST     HIGHLAND  34.88 503.4 -.824 .677
7.71  40.23 1
.22.74.84.811.11.1.77.841.01.41.61.91.51.21.21.71.81.71.21.1.94.45.58.29
.91.91.91.91.91.91.91.91.81.81.81.81.81.81.81.90.90.90.90.90.90.90.90
9.8   15.0  7.5   .065  .84   20.4  1.0   0.0
9.4   15.0  7.5   .065  .84   21.2  1.0   0.0
8.8   25.0  7.5   .065  .84   22.7  1.0   0.0
4.4   15.0  7.5   .065  .84   22.7  1.0   0.0

7.71  45.08 1
.75.971.1.94.78.811.11.01.2.881.31.21.71.41.11.11.21.3.71.68.45.36.19.23
1.01.01.01.01.01.01.01.0.93.93.93.93.93.93.93.931.11.11.11.11.11.11.11.1
12.9  15.0  7.5   .065  .84   15.5  1.0   0.0
9.4   15.0  7.5   .065  .84   21.2  1.0   0.0
9.3   25.0  7.5   .065  .84   21.4  1.0   0.0
3.8   15.0  7.5   .065  .84   26.3  1.0   0.0

7.71  43.91 1
.88.62.65.841.1.97.911.11.71.11.31.41.31.7.941.0.84.81.45.52.39.29.19.49
1.11.11.11.11.11.11.11.1.97.97.97.97.97.97.97.971.21.21.21.21.21.21.21.2
13.4  15.0  7.5   .065  .84   14.9  1.0   0.0
11.5  15.0  7.5   .065  .84   17.4  1.0   0.0
9.5   25.0  7.5   .065  .84   21.0  1.0   0.0
5.4   15.0  7.5   .065  .84   18.5  1.0   0.0

7.71  45.62 1
1.2.84.58.78.881.3.681.0.941.21.31.51.31.61.11.2.97.55.65.42.23.06.23.36
1.01.01.01.01.01.01.01.0.77.77.77.77.77.77.77.771.11.11.11.11.11.11.11.1
14.2  15.0  7.5   .065  .84   14.1  1.0   0.0
11.9  15.0  7.5   .065  .84   16.8  1.0   0.0
9.8   25.0  7.5   .065  .84   20.4  1.0   0.0
4.1   15.0  7.5   .065  .84   24.4  1.0   0.0
```

```
7.71  41.19 1
1.1.97.97.971.2.941.11.11.31.91.51.41.41.31.51.2.97.97.97.68.26.36.32.68
1.11.11.11.11.11.11.11.1.93.93.93.93.93.93.93.931.11.11.11.11.11.11.11.1
12.9  15.0  7.5   .065  .84   15.5  1.0   0.0
11.6  15.0  7.5   .065  .84   17.2  1.0   0.0
9.8   25.0  7.5   .065  .84   20.4  1.0   0.0
5.4   15.0  7.5   .065  .84   18.5  1.0   0.0

7.71  43.92 1
.71.58.65.71.75.781.01.2.911.71.21.51.42.12.21.2.91.75.75.49.32.29.36.42
.97.97.97.97.97.97.97.97.93.93.93.93.93.93.93.931.21.21.21.21.21.21.21.2
12.1  15.0  7.5   .065  .84   16.5  1.0   0.0
11.6  15.0  7.5   .065  .84   17.2  1.0   0.0
10.6  25.0  7.5   .065  .84   18.9  1.0   0.0
5.5   15.0  7.5   .065  .84   18.2  1.0   0.0

7.71  37.91 1
1.0.911.01.01.51.41.21.31.61.61.51.81.51.51.61.51.91.51.2.63.32.48.55.62
1.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.0
11.8  15.0  7.5   .065  .84   16.9  1.0   0.0
10.6  15.0  7.5   .065  .84   18.9  1.0   0.0
8.8   25.0  7.5   .065  .84   22.7  1.0   0.0
5.8   15.0  7.5   .065  .84   17.2  1.0   0.0

WEST      LOWLAND    51.92 678.5 -.476 .637
6.65  37.88 1
.38.60.94.53.831.11.31.31.51.71.41.41.71.81.51.92.31.5.86.90.49.53.23.38
0.90.90.90.90.90.90.90.90.90.90.90.90.90.90.90.91.11.11.11.11.11.11.11.1
10.1  15.0  7.5   .061  .81    9.9  1.0   0.0
9.4   15.0  7.5   .061  .81   10.6  1.0   0.0
8.6   25.0  7.5   .061  .81   11.6  1.0   0.0
5.3   15.0  7.5   .061  .81   18.9  1.0   0.0

6.65  39.51 1
.98.94.79.98.98.86.86.681.41.31.31.51.21.51.5.861.3.60.75.41.15.41.34.60
1.21.21.21.21.21.21.21.20.90.90.90.90.90.90.90.91.01.01.01.01.01.01.01.0
11.1  15.0  7.5   .061  .81    9.0  1.0   0.0
11.6  15.0  7.5   .061  .81    8.6  1.0   0.0
7.1   25.0  7.5   .061  .81   14.1  1.0   0.0
4.9   15.0  7.5   .061  .81   20.4  1.0   0.0

6.65  45.13 1
.711.2.68.49.641.41.11.51.21.41.41.61.41.11.51.3.981.1.26.23.23.15.53.64
0.80.80.80.80.80.80.80.81.11.11.11.11.11.11.11.11.11.11.11.11.11.11.11.1
10.7  15.0  7.5   .061  .81    9.3  1.0   0.0
10.3  15.0  7.5   .061  .81    9.7  1.0   0.0
9.1   25.0  7.5   .061  .81   11.0  1.0   0.0
4.9   15.0  7.5   .061  .81   20.4  1.0   0.0
```



```
6.65  37.41 1
.83.68.68.94.981.21.0.791.41.51.51.31.41.21.41.4.79.53.53.53.23.34.53.64
0.90.90.90.90.90.90.90.91.01.01.01.01.01.01.01.01.01.21.21.21.21.21.21.2
10.8  15.0  7.5   .061  .81    9.3  1.0   0.0
10.1  15.0  7.5   .061  .81    9.9  1.0   0.0
8.8   25.0  7.5   .061  .81   11.4  1.0   0.0
4.9   15.0  7.5   .061  .81   20.4  1.0   0.0

6.65  38.01 1
.411.11.0.681.2.90.981.01.21.31.61.61.21.61.31.1.64.60.34.71.38.26.38.64
1.01.01.01.01.01.01.01.00.90.90.90.90.90.90.90.91.31.31.31.31.31.31.31.3
12.1  15.0  7.5   .061  .81    8.3  1.0   0.0
11.1  15.0  7.5   .061  .81    9.0  1.0   0.0
8.0   25.0  7.5   .061  .81   12.5  1.0   0.0
5.9   15.0  7.5   .061  .81   16.9  1.0   0.0

6.65  42.21 1
.711.1.94.64.641.1.79.71.981.41.21.31.71.82.31.7.79.68.45.49.41.34.23.34
1.01.01.01.01.01.01.01.00.90.90.90.90.90.90.90.91.31.31.31.31.31.31.31.3
11.1  15.0  7.5   .061  .81    9.0  1.0   0.0
11.0  15.0  7.5   .061  .81    9.0  1.0   0.0
7.6   25.0  7.5   .061  .81   13.2  1.0   0.0
5.9   15.0  7.5   .061  .81   16.9  1.0   0.0

6.65  36.18 1
1.1.831.1.86.98.791.4.791.21.61.71.12.02.02.41.72.41.81.5.64.45.49.26.45
1.21.21.21.21.21.21.21.20.90.90.90.90.90.90.90.91.01.01.01.01.01.01.01.0
9.2   15.0  7.5   .061  .81   10.9  1.0   0.0
10.5  15.0  7.5   .061  .81    9.5  1.0   0.0
7.5   25.0  7.5   .061  .81   13.3  1.0   0.0
5.7   15.0  7.5   .061  .81   17.5  1.0   0.0

NORTH     HIGHLAND  24.08 409.1 -.997 .806
5.32  32.34 1
.94.77.851.31.31.01.3.991.61.41.51.81.61.41.11.72.01.01.3.42.61.47.19.61
1.11.11.11.11.11.11.11.1.83.83.83.83.83.83.831.11.11.11.11.11.11.11.1
7.5   15.0  7.5   .061  .85   13.3  1.0   0.0
6.3   15.0  7.5   .061  .85   15.9  1.0   0.0
6.5   25.0  7.5   .061  .85   15.4  1.0   0.0
3.1   15.0  7.5   .061  .85   32.2  1.0   0.0

5.32  35.94 1
1.31.21.3.801.5.94.991.01.11.51.61.0.901.11.11.51.1.71.56.52.42.24.38.85
1.11.11.11.11.11.11.11.1.90.90.90.90.90.90.90.901.01.01.01.01.01.01.01.0
10.3  15.0  7.5   .061  .85   10.0  1.0   0.0
7.8   15.0  7.5   .061  .85   15.0  1.0   0.0
6.0   25.0  7.5   .061  .85   16.0  1.0   0.0
3.0   15.0  7.5   .061  .85   33.3  1.0   0.0
```

```
5.32  34.26 1
.85.951.0.91.94.95.95.941.01.41.21.61.01.31.41.31.3.80.42.33.38.52.28.61
1.21.21.21.21.21.21.21.2.89.89.89.89.89.89.89.891.01.01.01.01.01.01.01.0
11.9   15.0  7.5    .061  .85    8.4  1.0   0.0
7.8    15.0  7.5    .061  .85   13.0  1.0   0.0
7.1    25.0  7.5    .061  .85   14.0  1.0   0.0
3.0    15.0  7.5    .061  .85   33.3  1.0   0.0

5.32  40.83 1
.75.94.94.66.71.861.31.31.21.11.31.31.51.01.41.6.801.2.66.38.28.28.33.71
1.01.01.01.01.01.01.01.0.90.90.90.90.90.90.90.901.21.21.21.21.21.21.21.2
11.3   15.0  7.5    .061  .85    9.0  1.0   0.0
6.5    15.0  7.5    .061  .85   16.0  1.0   0.0
7.8    25.0  7.5    .061  .85   13.0  1.0   0.0
4.1    15.0  7.5    .061  .85   25.0  1.0   0.0

5.32  34.99 1
1.31.6.71.89.89.85.891.41.31.41.51.61.71.1.85.89.71.85.52.66.19.19.33.52
1.11.11.11.11.11.11.11.11.31.31.31.31.31.31.31.01.01.01.01.01.01.01.0
12.0   15.0  7.5    .061  .85    8.0  1.0   0.0
7.1    15.0  7.5    .061  .85   14.0  1.0   0.0
6.5    25.0  7.5    .061  .85   16.0  1.0   0.0
3.8    15.0  7.5    .061  .85   26.0  1.0   0.0

5.32  37.10 1
.801.0.86.76.66.99.94.991.31.61.21.41.81.92.1.94.66.80.89.28.28.24.38.38
1.11.11.11.11.11.11.11.1.91.91.91.91.91.91.91.911.21.21.21.21.21.21.21.2
11.3   15.0  7.5    .061  .85    9.0  1.0   0.0
7.8    15.0  7.5    .061  .85   13.0  1.0   0.0
6.6    25.0  7.5    .061  .85   16.0  1.0   0.0
4.8    15.0  7.5    .061  .85   20.0  1.0   0.0

5.32  35.28 1
.661.11.2.85.85.80.661.01.51.01.01.7.941.61.72.32.21.21.6.90.91.28.28.28
1.21.21.21.21.21.21.21.2.89.89.89.89.89.89.89.89.97.97.97.97.97.97.97.97
7.5    15.0  7.5    .061  .85   13.0  1.0   0.0
7.9    15.0  7.5    .061  .85   12.5  1.0   0.0
6.3    25.0  7.5    .061  .85   16.0  1.0   0.0
4.8    15.0  7.5    .061  .85   20.0  1.0   0.0

SOUTH     LOWLAND    61.16 571.3 -.916 .773
5.50  38.85 1
.50.68.681.21.51.41.11.41.51.41.91.51.41.52.01.71.71.6.911.1.45.36.36.68
1.81.81.81.81.81.81.81.8.71.71.71.71.71.71.71.711.21.21.21.21.21.21.21.2
10.6   15.0  7.5    .067  .84    9.0  1.0   0.0
7.6    15.0  7.5    .067  .84   13.0  1.0   0.0
7.3    25.0  7.5    .067  .84   14.0  1.0   0.0
2.3    15.0  7.5    .067  .84   43.4  1.0   0.0
```



```
5.50  47.43 1
1.2.641.2.821.11.01.0.771.11.41.61.21.61.4.96.96.86.86.41.32.32.32.32.64
1.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.0
11.0   15.0  7.5    .067  .84    9.0  1.0   0.0
7.6    15.0  7.5    .067  .84   13.0  1.0   0.0
7.1    25.0  7.5    .067  .84   14.0  1.0   0.0
1.3    15.0  7.5    .067  .84   77.0  1.0   0.0

5.50  45.71 1
.59.77.86.91.91.961.11.41.61.61.91.71.21.51.81.1.91.36.18.18.05.18.32.59
1.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.0
12.6   15.0  7.5    .067  .84    8.0  1.0   0.0
6.9    15.0  7.5    .067  .84   14.0  1.0   0.0
6.8    25.0  7.5    .067  .84   15.0  1.0   0.0
1.8    15.0  7.5    .067  .84   55.5  1.0   0.0

5.50  44.77 1
.86.96.50.36.50.45.821.11.21.71.51.51.51.91.6.86.73.64.36.32.05.14.27.64
1.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.0
11.8   15.0  7.5    .067  .84    8.5  1.0   0.0
8.3    15.0  7.5    .067  .84   12.0  1.0   0.0
6.6    25.0  7.5    .067  .84   15.0  1.0   0.0
1.8    15.0  7.5    .067  .84   55.5  1.0   0.0

5.50  41.55 1
1.0.911.1.771.2.73.96.961.31.31.51.7.961.41.0.68.82.41.27.32.09.18.32.68
1.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.0
12.3   15.0  7.5    .067  .84    8.0  1.0   0.0
8.3    15.0  7.5    .067  .84   12.0  1.0   0.0
7.2    15.0  7.5    .067  .84   14.0  1.0   0.0
2.1    15.0  7.5    .067  .84   48.0  1.0   0.0

5.50  40.69 1
1.1.73.91.77.73.861.01.01.11.51.01.72.11.62.01.5.96.55.36.14.23.18.41.59
1.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.0
11.1   15.0  7.5    .067  .84    9.0  1.0   0.0
8.8    15.0  7.5    .067  .84   11.0  1.0   0.0
7.6    25.0  7.5    .067  .84   13.2  1.0   0.0
4.3    15.0  7.5    .067  .84   23.2  1.0   0.0

5.50  40.79 1
1.1.77.911.11.01.31.51.11.71.61.61.61.81.62.02.02.02.01.4.73.68.36.18.50
1.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.0
9.8    15.0  7.5    .067  .84   10.0  1.0   0.0
8.3    15.0  7.5    .067  .84   12.0  1.0   0.0
8.6    25.0  7.5    .067  .84   11.0  1.0   0.0
4.3    15.0  7.5    .067  .84   23.3  1.0   0.0
```

```
CENTRAL  LOWLAND   47.27 498.3 -.746 .658
3.66  37.31 1
.21.96.55.96.751.01.41.22.11.41.81.41.7.892.12.32.31.4.681.0.62.41.62.41
1.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.0
6.3    15.0  7.5    .067  .81    15.9  1.0    0.0
5.4    15.0  7.5    .067  .81    18.5  1.0    0.0
3.8    25.0  7.5    .067  .81    26.3  1.0    0.0
2.3    15.0  7.5    .067  .81    43.4  1.0    0.0

3.66  39.90 1
.68.681.51.21.3.891.0.89.961.71.51.21.61.21.11.3.96.55.55.34.20.21.62.27
0.90.90.90.90.90.90.90.91.01.01.01.01.01.01.01.01.01.01.01.01.01.01.01.0
6.8    15.0  7.5    .067  .81    14.7  1.0    0.0
5.0    15.0  7.5    .067  .81    20.0  1.0    0.0
3.5    25.0  7.5    .067  .81    28.5  1.0    0.0
2.8    15.0  7.5    .067  .81    35.7  1.0    0.0

3.66  40.20 1
.75.751.0.34.62.481.31.11.21.81.3.821.11.61.21.0.68.68.34.75.34.41.62.62
1.11.11.11.11.11.11.11.10.90.90.90.90.90.90.90.90.90.90.90.90.90.90.90.9
7.6    15.0  7.5    .067  .81    13.2  1.0    0.0
5.8    15.0  7.5    .067  .81    17.2  1.0    0.0
4.3    25.0  7.5    .067  .81    23.2  1.0    0.0
3.5    15.0  7.5    .067  .81    28.5  1.0    0.0

3.66  37.44 1
1.41.3.68.891.4.82.961.11.51.51.51.81.21.22.01.0.82.48.82.21.27.34.07.34
1.11.11.11.11.11.11.11.10.90.90.90.90.90.90.90.91.21.21.21.21.21.21.21.2
8.3    15.0  7.5    .067  .81    12.0  1.0    0.0
5.6    15.0  7.5    .067  .81    18.0  1.0    0.0
4.3    25.0  7.5    .067  .81    23.2  1.0    0.0
3.3    15.0  7.5    .067  .81    30.3  1.0    0.0

3.66  43.19 1
.621.2.68.89.68.75.621.11.31.41.42.11.62.01.1.82.21.41.27.48.14.27.21.41
1.01.01.01.01.01.01.01.0.90.90.90.90.90.90.90.901.41.41.41.41.41.41.41.4
6.6    15.0  7.5    .067  .81    15.0  1.0    0.0
5.3    15.0  7.5    .067  .81    18.9  1.0    0.0
3.8    25.0  7.5    .067  .81    26.3  1.0    0.0
3.1    15.0  7.5    .067  .81    32.2  1.0    0.0

3.66  39.66 1
.62.68.82.55.89.98.891.31.81.61.11.62.11.81.61.7.62.41.41.62.27.07.41.55
1.01.01.01.01.01.01.01.0.90.90.90.90.90.90.90.901.31.31.31.31.31.31.31.3
7.1    15.0  7.5    .067  .81    14.0  1.0    0.0
6.9    15.0  7.5    .067  .81    14.0  1.0    0.0
3.5    25.0  7.5    .067  .81    28.5  1.0    0.0
4.1    15.0  7.5    .067  .81    25.0  1.0    0.0

3.66  38.60 1
.27.821.31.3.681.21.0.891.21.71.1.961.62.32.02.52.31.81.41.1.96.20.34.14
1.11.11.11.11.11.11.11.11.01.01.0
```


Appendix B

PROGRAM FOR ESTIMATING THE RELATIONSHIP BETWEEN CFS UNAVAILABILITIES AND NON-CFS UNAVAILABILITIES[1]

INTRODUCTION

Research conducted by students at UCLA on the allocation process for the Los Angeles Police Department[2] found that unavailabilities of patrol cars for reasons other than calls for service (because of traffic enforcement, meal times, and the like) vary according to the cfs workload of the cars. These unavailabilities, which are an important component for patrol car activity in all police departments, reduce the effective number of cars available to service calls. Scheduled unavailabilities such as meals, and unscheduled but predictable activities such as automobile service stops, can be expected to occur independently of the cfs workload. However, discretionary officer-initiated activity might increase during slack periods and decrease during overloaded periods of the day, or vice versa.

In Los Angeles, the variation was found to be rather complicated. Data collected at the dispatch center showed that more time was spent per car on non-cfs unavailabilities when the cfs workload was high than when it was low. An apparent explanation for this is that the times of day in which many emergencies are reported to the police by telephone also have many activities visible from the street that require police intervention.

However, the queuing delays experienced before a car could be dispatched to a call in Los Angeles indicated that more cars were unavailable at time of low cfs workload than were reported unavailable. Moreover, it sometimes happened at times of high cfs workload that more cars were actually available than were reported available. Extra patrol

---

[1]David J. Jaquette, who also wrote the program described here, was coauthor of this appendix.

[2]*An Analysis of the Patrol Car Deployment Methods of the Los Angeles Police Department*, Engineering School report by Public Systems Analysis class, University of California at Los Angeles, 1975.

units, such as traffic cars and sergeants' cars, could be used to handle cfs work if necessary.

Because the equations in a patrol allocation program should be designed to predict queuing delays as they will actually occur, it is appropriate to estimate the effective number of patrol cars present in the field from data giving the number of calls delayed, and not from data telling how many cars were fielded and their reported unavailabilities.

Thus if NEFF denotes the effective number of patrol cars that, according to queuing formulas, would cause the observed fraction of calls delayed, our estimate of the fraction of time each car is unavailable (UNAVL) on non-cfs activity is

$$\text{UNAVL} = 1 - \frac{\text{NEFF}}{\text{CARS}}, \qquad \text{(B.1)}$$

where CARS is the number of cars fielded.

Then, the effective fraction of time unavailable (UNAVL) is modeled to be linearly related to the fraction of time the average car spends on calls for service, C:

$$\text{UNAVL} = B1 \times C + B2, \qquad \text{(B.2)}$$

where B1 and B2 are coefficients specific to each precinct but assumed to be time-homogeneous. This relationship was found to explain the relationship between effective and fielded cars in Los Angeles, as evidenced by the increase in the number of calls delayed during slack periods and the decrease during periods of heavy demand in calls for service. (See Fig. 3 in the User's Manual.)

Given this relationship, the only input data related to non-cfs unavailabilities needed by PCAM is the pair of unavailability parameters B1 and B2 for each precinct. The computer program listed and annotated here was written originally as an aid to the LAPD Automated Deployment of Available Manpower (ADAM) project in their attempt to implement a version of PCAM. It can be used to construct estimates of the unavailability parameters.

INPUT DATA

The program takes raw data from LAPD records and converts them into numbers usable in a standard linear regression. Each data point read in on one data card represents a number of weeks (NWEEKS) of aggregated data for one shift. For example, one line of printout in the data summary available to the LAPD would describe the activity of patrol cars in the Van Nuys area during the tour from midnight to 3 a.m. on Mondays, over a four-week period; thus NWEEKS = 4. Each input card contains the total number of actual car-hours (AVLHR), which in this example would be 4 × 3 times the average number of cars fielded; the number of hours in the shift (NHOURS), which in the example is 3; actual hours spent on calls-for-service work (CFSWRK); total number of delayed calls (NDELAY); and total calls for service (NTCFS).

The fraction of each car's time spent on calls for service, which is the independent variable of the regression, is immediately found as the calls-for-service workload divided by the total actual car-hours, NH

$$C = \frac{\text{CFSWRK}}{\text{AVLHR}}.$$

This is calculated for each shift.

For each shift, dividing the number of calls delayed by the total number of calls gives the fraction delayed, which is an estimate of the probability of delay. If there are N effective cars on duty, and the number of cfs work hours per hour is ρ, the formula for an M/M/N queue shows that the probability of a call being delayed is

$$P(\text{delay}|N) = \frac{\rho N/(N!(1 - \rho/N))}{1 + \rho + \rho^2/2! + \ldots + \rho^{(N-1)}/(N-1)! + \rho^N/N!(1 - \rho/N)}. \qquad \text{(B.3)}$$

A maximum-likelihood and unbiased estimate of the number of *effective* cars during a given shift can be made by solving N in the relationship

$$P(\text{delay}|N) = \text{actual fraction of calls delayed.}$$

The value ρ needed in the above calculation is found as the actual calls-for-service workload hours (CFSWRK) divided by the number of total hours contained in the data for that shift (NWEEKS × NHOURS):

$$\rho = \frac{\text{CFSWRK}}{\text{NWEEKS} \times \text{NHOURS}}.$$

Once ρ and the fraction delayed are estimated, N can be determined by evaluating the expression above for a K such that

$$P(\text{delay}|K) > \text{actual fraction of calls delayed}$$

and

$$P(\text{delay}|K + 1) < \text{actual fraction of calls delayed.}$$

Linear interpolation between K and K + 1 is used to estimate N. The ratio of N to the actual number of cars fielded, CARS, gives an estimate of the fraction of time unavailable, UNAVL, as shown in Eq. (B.1), (CARS = AVLHRS/NWEEKS × NHOURS).

Once UNAVL and C have been calculated for each shift in a precinct, the usual formulas for a regression fit are used to estimate B1 and B2 for that precinct:

$$B1 = \frac{n \Sigma \text{UNAVL}_i C_i - \Sigma \text{UNAVL}_i \Sigma C_i}{n \Sigma C^2_i - (\Sigma C_i)^2},$$

where n is the number of observations, and

$$B2 = \Sigma(\text{UNAVL}_i - B1 \times C_i)/n.$$

INPUT DATA FORMAT FOR PROGRAM TO CALCULATE B1 AND B2

The format instructions may be clarified by the sample data file that follows.

1. *Control card.* Enter the number of precincts for which data are provided in columns 1-2, format I2.
2. *Cards for each precinct.*
   a. *Precinct name.* Enter precinct name on one card, left justified.
   b. *Number of data cards for this precinct.* Enter on one card in columns 1-2, format I2.
   c. *Data cards.* One for each shift.

Position	*Format*	*Description*
1-10	F10.1	AVLHR Number of actual car-hours fielded in the shift
11-12	I2	NHOURS Number of hours in the shift
13-22	F10.2	CFSWRK Number of car-hours of cfs work
23-25	I3	NDELAY Number of calls delayed
26-28	I3	NTCFS Total number of calls for service

Sample Data File for Program To Calculate B1 and B2

Note that an error has been purposely introduced for the first shift in WEST precinct. The number of calls delayed (64) exceeds the total number of calls (63). This data card (observation 1 in WEST precinct) will be ignored by the program.

```
Column 1
|
 3
WEST
 5
111.          3 85.2      64 63
140.          5 31.        4 23
312.          8 151.4     67113
123.          3 71.9      46 54
240.          5 104.      40 78
DOWNTOWN
 5
105.          3 42.3      12 31
140.          5 30.5       7 25
336.          8 189.7    110142
126.          3 106.3     82 95
245.          5 130.9     65. 98
NORTH
 7
        122   3 38.3      11 34
        152   5 25.3       4 28
        362   8 195.4    103155
        138   3 102.4     84 92
        210   5 118.2     58 95
        128   3 42.3      13 36
        158   5 32.4       5 30
```

Output from Running Program with Sample Data File

```
ERROR IN DELAY DATA FOR OBS  1 IN WEST          PRECINCT

 FOR WEST          PRECINCT  B1=   -0.3476 B2=     0.5876

 FOR DOWNTOWN      PRECINCT  B1=   -0.7170 B2=     0.7476

 FOR NORTH         PRECINCT  B1=   -0.6336 B2=     0.7015
```


LISTING OF PROGRAM TO CALCULATE UNAVAILABILITY PARAMETERS B1 AND B2

```
      DIMENSION PROB(20)
      INTEGER FACTN(41)
      DATA NWEEKS/4/
C
C
C  CALCULATE FACTORIAL(I) AS FACTN(I+1)
C
      FACTN(1) = 1.
      DO 1 I=2,40
1       FACTN(I) = FACTN(I-1)*(I-1)
C
      READ (5,101) NDIST
101   FORMAT(I2)
      DO 30 IJ=1,NDIST
       READ (5,102) PCTNM1,PCTNM2,PCTNM3
102    FORMAT(3A4)
       READ (5,103) NOBSV
103    FORMAT (I2)
       SUMY=0.0
       SUMYSQ=0.0
       SUMC=0.0
       SUMCSQ=0.0
       SUMYC=0.0
       NOBS = 0
       DO 20 IK=1,NOBSV
        READ (5,104) AVLHR,NHOURS,CFSWRK,NDELAY,NTCFS
        CARS= AVLHR/(NWEEKS*NHOURS)
        NCARS = CARS + .99999999
104     FORMAT(F10.1,1I2,F10.2,2I3)
        RHO= CFSWRK/(NHOURS*NWEEKS)
        DELAYP = NDELAY
        DELAYP=DELAYP/NTCFS
        IF (DELAYP.GT.1.0) GO TO 19
C
C
C  CALCULATE INTEGER N-EFFECTIVE FROM QUEUING FORMULA
C

        NEFF = 1
        LOWCAR= RHO + 1
        DO 5 I=LOWCAR,NCARS
         DENSUM= 1.
         ILESS1= I-1
         DO 4 IL= 1,ILESS1
4         DENSUM= DENSUM + RHO**IL/FACTN(IL+1)
         XNUM = RHO**I/((1.-RHO/I)*FACTN(I+1))
         PROB(I) = XNUM/(DENSUM+XNUM)
         NEFF = I
         IF (PROB(I).LEDELAYP) GO TO 11
```

```
5       CONTINUE
C
C
C  THE FOLLOWING IS AN INTERPOLATION FOR EFFECTIVE N
C  IF THE CLOSEST NEFFECTIVE CARS IS GREATER THAN OR EQUAL TO THE
C  ACTUAL NUMBER OF CARS, THEN THE INTERPOLATION IS BYPASSED, AND
C  THE TIME SPENT ON NON-CFS WORK IS SET TO ZERO
C
11      AEFF= NEFF
        IF (AEFF.GE.CARS) GO TO 7
        JK= NEFF - 1
        IF (NEFF.GT.LOWCAR) EFFN=(PROB(NEFF)-DELAYP)/
    1   (PROB(JK)-PROB(NEFF)) +AEFF
        IF (NEFF.EQ.LOWCAR) EFFN=(1.0-DELAYP)/
    1   (1.0-PROB(NEFF)) + RHO
        UNAVL = AMAX1(0.0,1-EFFN/CARS)
        GO TO 8
7       UNAVL = 0.0
        EFFN= CARS
C
C
C  ACCUMULATE TERMS FOR REGRESSION COEFFICIENTS
C
8       C = RHO/CARS
       SUMY= SUMY + UNAVL
       SUMYSQ= SUMYSQ + UNAVL*UNAVL
       SUMC = SUMC + C
       SUMCSQ = SUMCSQ + C*C
       SUMYC= SUMYC + UNAVL*C
       NOBS = NOBS+1
       GO TO 20
C
19     WRITE(6,123) IK,PCTNM1,PCTNM2,PCTNM3
123    FORMAT(' ERROR IN DELAY DATA FOR OBS',I4,
    1  ' IN ',3A4,' PRECINCT')
20     CONTINUE
C
C
C  CALCULATE REGRESSION COEFFICIENTS
C
         YC= NOBS*SUMYC-SUMY*SUMC
         CC= NOBS*SUMCSQ-SUMSQ-SUMC*SUMC
       B1= YC/CC
       B2= SUMY/NOBS - B1* SUMC/NOBS
C
       WRITE(6,106)  PCTNM1,PCTNM2,PCTNM3,B1,B2
106    FORMAT('0 FOR ',3A4,' PRECINCT  B1= ',F10.4,' B2= ',
    1  F10.4)
30      CONTINUE
       CALL EXIT
       END
```


Appendix C

PCAM REFERENCE SHEETS

GENERAL

AMPERSAND (&): At end of line, signifies command continues on the following line.

DELIMITER: Any character other than a letter, digit, parenthesis, hyphen, period, asterisk, or ampersand. Can be used freely to improve readability. Examples: blank, comma, colon, semicolon, equal sign.

FILLER WORD: FOR, CAR, CARS, HOUR, HOURS, TO, ON, BY, DATA. Ignored by program.

QUALIFIER: Any combination of
TOUR=<NAMELIST>
DIVISION=<NAMELIST>
PRECINCT=<NAMELIST>
DAY=<NAMELIST>
The words on the left of the equal sign are supplied by the user, except for DAY. The qualifier may be omitted in any command.

COMMANDS

1. HEADR [Run name]

   Reads the run name and stores it for printing at the top of all output reports.

2. READ [DATA] [FOR] <QUALIFIER>

   Reads data from DATABASE into CURRENT-DATA, establishes default output order for DISP, and increases number of cars assigned (if necessary) to assure that enough cars are on duty in every block to handle the call-for-service workload. The first command in any run of PCAM must be READ.

3. LIST [DATA] [FOR] <QUALIFIER>

   Lists data from CURRENT-DATA. Averages some data.

4. DISP T<NUMBERLIST> [FOR] <QUALIFIER>

   DISP A<NUMBERLIST> [FOR] <QUALIFIER>

   Displays the output tables specified in <NUMBERLIST>. <QUALIFIER> establishes the output order within each table as

well as the scope. In the A form of the command only the average and total lines are printed.

5. ALOC <NUMBER> [CAR] [HOURS] [TO] <QUALIFIER> [BY] F<NUMBERLIST>

   ALOC * [CAR] [HOURS] [TO] <QUALIFIER> [BY] F<NUMBERLIST>

   ALOC * -<NUMBER> [CAR] [HOURS] [TO] <QUALIFIER> [BY] F<NUMBERLIST>

   Allocates the specified number of car-hours so as to minimize F<NUMBERLIST>. Asterisk (*) represents the number currently assigned. At a minimum, allocates enough cars to handle the call-for-service workload in every block.

6. ADD <NUMBER> [CAR] [HOURS] [TO] <QUALIFIER> [BY] F<NUMBERLIST>

   ADD <NUMBER> -* [CAR] [HOURS] [TO] <QUALIFIER> [BY] F<NUMBERLIST>

   Adds the specified number of car-hours to the number currently assigned so as to minimize F<NUMBERLIST>. In the second version, execution of the command will result in the total number of car-hours assigned equaling <NUMBER>.

7. MEET C<NUMBERLIST>$_1$=<NUMBERLIST>$_2$ [FOR] <QUALIFIER>

   Assigns enough car-hours to each specified shift to assure that the measures indicated in <NUMBERLIST>$_1$ meet the constraints in <NUMBERLIST>$_2$ for every time block. One constraint value must be specified for each measure.

   If no ALOC, ADD, or MEET commands have been entered since the last READ command, MEET assigns the minimum number of car-hours needed to meet constraints and keep utilization of an effective car under 1 in every hour. Otherwise, car-hours are added to those already allocated, if needed to meet the constraints.

8. SET P<NUMBERLIST>$_1$=<NUMBERLIST>$_2$ [FOR] <QUALIFIER>

   Changes specified data items. There must be a one-to-one correspondence between data items in <NUMBERLIST>$_1$ and values in <NUMBERLIST>$_2$. SET also checks that enough car-hours are assigned to each shift so as to keep the utilization of an effective car under 1 in each hour.

9. WRITE [DATA] [ON] <NUMBER> [FOR] <QUALIFIER>

   Writes a NEW-DATA file on Fortran unit <NUMBER>. NEW-DATA contains the part of CURRENT-DATA specified by <QUALIFIER>.

10. END

    Terminates program. Must be last command.

OBJECTIVE FUNCTIONS FOR ALOC AND ADD

F(1) Average fraction of calls delayed in queue

F(2) Average length of time calls are delayed in queue

F(2,N) Average length of time priority N calls are delayed in queue

F(3) Average total response time (queuing + travel time)

CONSTRAINT SPECIFICATIONS FOR MEET[3]

C(1)	Percent of time an average car is busy handling calls for service	↓
C(2)	Average travel time (minutes)	↓
C(3)	Average number of cars available (same as average patrol hours per hour)	↑
C(5)	Patrol interval (minutes)	↓
C(6)	Minimum number of cars	↓
C(7)	Percent of calls delayed	↓
C(8)	Average delay, priority 2 (minutes)	↓
C(9)	Average delay, priority 3 (minutes)	↓
C(10)	Average delay for all calls (minutes)	↓
C(11)	Total delay (queuing + travel), priority 2 (minutes)	↓
C(12)	Total delay (queuing + travel), priority 3 (minutes)	↓
C(13)	Total delay (queuing + travel) for all calls (minutes)	↓

DATA ITEMS FOR SET

P(1)	Unavailability parameter B1	(precinct)
P(2)	Unavailability parameter B2	(precinct)
P(3)	Call rate parameter	(day, in precinct)
P(4)	Service time parameter	(day, in precinct)
P(5)	Actual cars assigned	(shift)
P(6)	Response speed	(shift)
P(7)	Patrol speed	(shift)
P(8)	Fraction of calls priority 1	(shift)
P(9)	Fraction of calls priority 2	(shift)
P(10)	Officers per car	(shift)
P(11)	Smoothing flag	(universal)

[3]The arrows indicate that the constraint is met if the measure is *lower* than the specified value for downward arrows (↓) or *higher* than the specified value for upward arrows (↑).

SAMPLE SEQUENCE OF COMMANDS

Command	Explanation
READ DATA FOR DIVISION=HIGHLAND	Data covering an entire week in Highland Division are read into CURRENT-DATA.
SET P(3)=7.85 FOR PRECINCT=EAST SET P(3)=5.45 FOR PRECINCT=NORTH	Call rates have increased slightly since the last time PCAM was used. These commands adjust the call rates.
DISP T(1,2,3,4,5)	User wants to see what has happened to performance measures with the new call rates.
ALOC 4000 CAR HOURS BY F(2) DISP T (3)	User wants to see how to allocate the number of car-hours (4000) now planned for this division to minimize average queuing delay.
ALOC * BY F(3) DISP T (3)	Morning tours (with fast travel speed) appear to have unnecessarily low response times in the previous allocation, while most tours are too high. User attempts to minimize average response time, but finds he gets nearly the same allocation.
ADD 128 BY F(3) DISP T (2)	User wants to see how much improvement can be obtained by allocating four more patrol officers (each performing patrol car duty 32 hours a week) to this division.
MEET C(10)=21	Response times are still too high. User wants to know how many additional car-hours are needed to keep response time under 21 minutes in every shift.
READ DIVISION=HIGHLAND SET P(3)=7.85 PRECINCT=EAST SET P(3)=5.45 PRECINCT=NORTH MEET C(10)=21	Number of car-hours needed in the previous allocation is too large. User starts over, trying to meet constraints *before* allocating. He uses the short form of the commands this time.

ADD 4000-* CAR HOURS BY F(3)	The user had some car-hours left over after meeting the constraint. Now he allocates the remainder so as to minimize average travel time.
DISP T(1,2,3,4,5)	The user examines the results of this allocation.
WRITE DATA ON 18	The user thinks this is a good allocation and writes out a NEW-DATA file.
END	User terminates this session with PCAM.

Appendix D

PROGRAM CROSS-REFERENCE TABLE

Symbol	Defined In	Referenced In
ADDALC	ADDALC	MAIN
ADDCAR	ADDCAR	ADDALC
ADJUST	ADJUST	ADDALC
AVTT	AVTT	KNSTR
BUFFER	BLKDAT	BLKDAT
BUFFER		GETTKN
BUFFER		HEAD
CEIL	CEIL	ADDALC
CEIL		DERIVE
CEIL		MEET
CKOVR	CKOVR	ADDALC
CKOVR		MEET
CKOVR		WRITE
COMPTB	COMPTB	DSPDTP
COMPTB		DSPPDT
CRLEFT	CRLEFT	OBJF1
CRLEFT		WLEFT
DERIVE	DERIVE	READ
DERIVE		SET
DISP	DISP	MAIN
DSPDTP	DSPDTP	DISP
DSPPDT	DSPPDT	DISP
GETBOT	GETBOT	INIT
GETBOT		READ
GETTKN	GETTKN	SCAN
GETTOP	GETTOP	ADDALC
GETTOP		SCAN
GETTOP		WRITE
GTDSPC	GTDSPC	ADDALC
GTDSPC		DISP
GTDSPC		LIST

Symbol	Defined In	Referenced In
GTDSPC		MEET
GTDSPC		READ
GTDSPC		SET
GTDSPC		WRITE
HEAD	HEAD	MAIN
INIT	INIT	MAIN
KEYWDS	BLKDAT	ADDALC
KEYWDS		BLKDAT
KEYWDS		DERIVE
KEYWDS		DISP
KEYWDS		DSPDTP
KEYWDS		DSPPDT
KEYWDS		GTDSPC
KEYWDS		INIT
KEYWDS		LIST
KEYWDS		MAIN
KEYWDS		MEET
KEYWDS		READ
KEYWDS		SCAN
KEYWDS		SET
KEYWDS		SETWFL
KEYWDS		WRITE
KNSTR	KNSTR	MEET
LCODES	BLKDAT	BLKDAT
LCODES		GETTKN
LCODES		SCAN
LIST	LIST	MAIN
LKP1	LKP1	GETTKN
LKP1		GTDSPC
LKP1		MRGORD
LKP1		READ
LKP8	LKP8	NXPCT
LKP8		READ
LKP8		SCAN
LKP8		SETWFL
MEET	MEET	MAIN
MOVE	MOVE	DSPDTP
MOVE		DSPPDT
MOVE		HEAD

Symbol	Defined In	Referenced In
MOVE		MRGORD
MOVE		READ
MOVE		SCAN
MOVE		WRITE
MRGORD	MRGORD	DISP
MRGORD		READ
NXDAY	NXDAY	ADDALC
NXDAY		ADDCAR
NXDAY		DSPDTP
NXDAY		DSPPDT
NXDAY		LIST
NXDAY		MEET
NXDAY		SET
NXDAY		WRITE
NXPCT	NXPCT	ADDALC
NXPCT		ADDCAR
NXPCT		DSPDTP
NXPCT		DSPPDT
NXPCT		LIST
NXPCT		MEET
NXPCT		SET
NXPCT		WRITE
NXTOUR	NXTOUR	ADDALC
NXTOUR		ADDCAR
NXTOUR		DSPDTP
NXTOUR		DSPPDT
NXTOUR		LIST
NXTOUR		MEET
NXTOUR		SET
NXTOUR		STRCAR
NXTOUR		WRITE
OBJFUN	OBJFUN	ADJUST
OBJFUN		SBLOBJ
OBJF1	OBJF1	COMPTB
OBJF1		KNSTR
OBJF1		OBJFUN
OBJF2	OBJF2	COMPTB
OBJF2		KNSTR
OBJF2		OBJFUN
OBJF2		OBJF3
OBJF3	OBJF3	COMPTB
OBJF3		KNSTR

Symbol	Defined In	Referenced In
OBJF3		OBJFUN
OFFSET	BLKDAT	ADDALC
OFFSET		ADDCAR
OFFSET		ADJUST
OFFSET		AVTT
OFFSET		BLKDAT
OFFSET		COMPTB
OFFSET		DERIVE
OFFSET		DSPDTP
OFFSET		DSPPDT
OFFSET		KNSTR
OFFSET		LIST
OFFSET		MEET
OFFSET		NXDAY
OFFSET		NXPCT
OFFSET		NXTOUR
OFFSET		OBJFUN
OFFSET		OBJF2
OFFSET		READ
OFFSET		SBLACT
OFFSET		SBLEF
OFFSET		SBLOBJ
OFFSET		SET
OFFSET		STRCAR
OFFSET		STRDF
OFFSET		STROBJ
OFFSET		WRITE
OPTION	BLKDAT	BLKDAT
OPTION		INIT
OPTION		READ
OPTION		WRITE
PDEL	PDEL	OBJF1
PDEL		OBJF2
PDEL		WLEFT
PNTRS	BLKDAT	ADDALC
PNTRS		ADDCAR
PNTRS		ADJUST
PNTRS		BLKDAT
PNTRS		CKOVR
PNTRS		COMPTB
PNTRS		DERIVE
PNTRS		DISP
PNTRS		DSPDTP
PNTRS		DSPPDT
PNTRS		GTDSPC

Symbol	Defined In	Referenced In
PNTRS		INIT
PNTRS		KNSTR
PNTRS		LIST
PNTRS		MEET
PNTRS		NXDAY
PNTRS		NXPCT
PNTRS		NXTOUR
PNTRS		OBJFUN
PNTRS		READ
PNTRS		SBLACT
PNTRS		SBLEF
PNTRS		SBLOBJ
PNTRS		SET
PNTRS		SETWFL
PNTRS		STRCAR
PNTRS		STRDF
PNTRS		STROBJ
PNTRS		WRITE
PRTBL	PRTBL	DSPDTP
PRTBL		DSPPDT
PRTBL		TOTAL
READ	READ	MAIN
SBLACT	SBLACT	ADDALC
SBLACT		DERIVE
SBLACT		MEET
SBLEF	SBLEF	ADDALC
SBLEF		DERIVE
SBLEF		MEET
SBLOBJ	SBLOBJ	ADDALC
SBLOBJ		ADDCAR
SBLOBJ		ADJUST
SCAN	SCAN	ADDALC
SCAN		DISP
SCAN		GTDSPC
SCAN		LIST
SCAN		MAIN
SCAN		MEET
SCAN		READ
SCAN		SET
SCAN		WRITE
SCODES	BLKDAT	ADDALC
SCODES		BLKDAT

Symbol	Defined In	Referenced In
SCODES		DISP
SCODES		GTDSPC
SCODES		LIST
SCODES		MAIN
SCODES		MEET
SCODES		READ
SCODES		SCAN
SCODES		SET
SCODES		WRITE
SET	SET	MAIN
SETWFL	SETWFL	ADDALC
SETWFL		DISP
SETWFL		LIST
SETWFL		MEET
SETWFL		SET
SETWFL		WRITE
SKIP	SKIP	READ
STATS	BLKDAT	BLKDAT
STATS		COMPTB
STATS		DISP
STATS		READ
STATS		TOTAL
STATS		ZERO
STORE	BLKDAT	ADDALC
STORE		ADDCAR
STORE		ADJUST
STORE		AVTT
STORE		BLKDAT
STORE		CKOVR
STORE		COMPTB
STORE		DERIVE
STORE		DISP
STORE		DSPDTP
STORE		DSPPDT
STORE		GETBOT
STORE		GETTOP
STORE		INIT
STORE		KNSTR
STORE		LIST
STORE		MAIN
STORE		MEET
STORE		NXDAY
STORE		NXPCT
STORE		NXTOUR
STORE		OBJFUN

Symbol	Defined In	Referenced In
STORE		OBJF1
STORE		OBJF2
STORE		PDEL
STORE		READ
STORE		SBLACT
STORE		SBLEF
STORE		SBLOBJ
STORE		SCAN
STORE		SET
STORE		SETWFL
STORE		STRCAR
STORE		STRDF
STORE		STROBJ
STORE		WLEFT
STORE		WRITE
STRCAR	STRCAR	ADDALC
STRCAR		MEET
STRDF	STRDF	ADDCAR
STRDF		STROBJ
STROBJ	STROBJ	ADDALC
STROBJ		ADDCAR
STROBJ		ADJUST
SYSTEM	BLKDAT	ADDALC
SYSTEM		ADDCAR
SYSTEM		BLKDAT
SYSTEM		CKOVR
SYSTEM		DERIVE
SYSTEM		DISP
SYSTEM		DSPDTP
SYSTEM		DSPPDT
SYSTEM		GETBOT
SYSTEM		GETTKN
SYSTEM		GETTOP
SYSTEM		GTDSPC
SYSTEM		INIT
SYSTEM		LIST
SYSTEM		MAIN
SYSTEM		MEET
SYSTEM		OBJF2
SYSTEM		READ
SYSTEM		SCAN
SYSTEM		SET
SYSTEM		SETWFL
SYSTEM		TITLE
SYSTEM		TOTAL
SYSTEM		WRITE

Symbol	Defined In	Referenced In
TITLE	TITLE	DSPDTP
TITLE		DSPPDT
TITLES	BLKDAT	BLKDAT
TITLES		DISP
TITLES		HEAD
TITLES		INIT
TITLES		LIST
TITLES		READ
TITLES		TITLE
TITLES		WRITE
TOTAL	TOTAL	DSPDTP
TOTAL		DSPPDT
TRAVEL	TRIDSP	OBJF1
TRAVEL		OBJF2
TRAVEL		OBJF3
TRAVEL		TRIDSP
TRAVEL		WLEFT
TRIDSP	TRIDSP	CRLEFT
TRIDSP		DERIVE
TRIDSP		OBJF1
TRIDSP		OBJF2
TRIDSP		PDEL
TRIDSP		WLEFT
WLEFT	WLEFT	OBJF2
WRITE	WRITE	MAIN
ZERO	ZERO	DSPDTP
ZERO		DSPPDT

Appendix E

ADDRESSES FOR FURTHER INFORMATION

1. For copies of the PCAM program on card or tape, answers to questions about the program, and information about related emergency service deployment models:

Dr. Warren E. Walker
The Rand Corporation
P.O. Box 2138
Santa Monica, California 90406-2138
(213) 393-0411

Dr. Jan M. Chaiken
55 Wheeler Street
Cambridge, Massachusetts 02138
(617) 492-7100

2. Research sponsor

National Institute of Justice

Dr. George Shollenberger
National Institute of Justice
Office of Evaluation
633 Indiana Avenue, N.W.
Washington, D.C. 20530
(202) 376-3933

REFERENCES

1. Jan M. Chaiken and Peter Dormont, *Patrol Car Allocation Model: User's Manual,* The Rand Corporation, R-1786/2-HUD/DOJ, Santa Monica, 1975.

2. Linda Green, "A Multiple Dispatch Queueing Model of Police Patrol Operations," *Management Science,* Vol. 30, 1984, pp. 653-664.

END

CONTINUED

3 OF 3