who use technology and those who create it. On a worldwide scale, leaders of corporations and countries are struggling to understand the Y2K problem. In the process, they are receiving a crash course in the fragile mechanics of information technology.

The Committee feels strongly that Y2K, as the first widespread challenge of the information age, must leave a legacy of increased awareness and appreciation of information technology's role in social and economic advancement.

UNDERSTANDING THE PROBLEM

The goal of this section is to provide background on the Y2K problem and answer common Y2K questions.

What is the Year 2000 computer technology problem?

The phrases the "Year 2000 Computer Technology Problem," the "Millennium Bug," the "Century Date Change," or simply, "Y2K"[1] all refer to the same problem—a defect that exists in millions of computer programs worldwide that causes erroneous handling of date (i.e., day, month and year) information if not corrected. The effect of the Y2K flaw on computer systems is not easily predictable. It may bring a computer to a crashing halt. It may cause the computer to generate obviously incorrect outputs. Or alternatively, it could allow the computer to produce invalid data that will not be detected until much later, forcing users to correct a range of accumulated errors while searching for the source of the problem.

Why is two digit notation defective?

To save memory in the early days of computing, programmers represented four-digit years with only two digits. For instance, 1968 or 1974 would be stored and processed as 68 and 74, respectively. The number 19, indicating years in the 1900s, was implied, much as personal checks once had the number 19 preprinted on the dateline.

This worked smoothly until users started to input dates occurring after December 31, 1999. Computers ran into problems when required to calculate a number based on the difference in two dates, such as the interest due on a mortgage loan. Computers continued to assume that the prefix 19 was implied, so dates such as 00 or 01 were treated as 1900 or 1901. Consequently, computers could not correctly calculate the difference between a year in the 20th century and a year in the 21st century.

For example, we know that the time between July 1, 1998, and July 1, 2005 is exactly 7 years. However, a computer with a Y2K problem could calculate an answer of either 93 years or -7 years, depending on the specific program. Calculations that used either of these results would be in error and may themselves cause subsequent problems.

Another Y2K problem occurs in the storage of information. Many kinds

of data are organized and processed by date, such as driver's license records and credit card accounts. Computers have had problems processing credit cards that have expiration dates after December 1999. Due to two-digit dating, computers have thought that cards expiring in 2000 or later had expired almost a century ago.

What is the scope of Y2K problems?

The Y2K problem affects two general classes of equipment. The first class comprises business systems or mainframe systems. These computers perform a variety of data-intensive calculations—balancing accounts, making payments, tracking inventory, ordering goods, managing personnel, scheduling resources, etc. The second class of equipment has several common names, including embedded chips, embedded processors and embedded control systems. Many aspects of modern society rely on microchip-enhanced technology to control or augment operations. Examples are ubiquitous. Automatic teller machines, toll collection systems, security and fire detection systems, oil and gas pipelines, consumer electronics, transportation vehicles, manufacturing process controllers, military systems, medical devices and telecommunications equipment all depend on embedded chip-technology.

Y2K related failures in business systems will generally cause an enterprise to lose partial or complete control of critical processes. In the private sector, loss of business systems means that a company may have difficulty managing its finances, making or receiving payments and tracking inventory, orders, production or deliveries. In the public sector, government organizations may be severely hindered in performing basic functions such as paying retirement and medical benefits, maintaining military readiness, responding to state and local emergencies, controlling air traffic, collecting taxes and customs and coordinating law enforcement efforts.

Y2K problems in embedded systems have the potential to affect public health and safety. Problems that need to be fixed have already been detected in medical treatment devices, water and electricity distribution and control systems, airport runway lighting and building security systems.[2] Other suspect areas are pipeline control systems and chemical and pharmaceutical manufacturing processes.

How was the Y2K mistake made?

Several factors explain the creation of the Y2K problem. In the early days of computers, computer memory was very expensive. In the IBM 7094 of the early 1960s, core memory cost around $1 per byte. Today's semiconductor memory costs around $1 per million bytes. Thus, there was a very strong economic incentive to minimize the amount of memory needed to store a program and its data in the computer's memory.

Additionally, early computer programming was highly time-consuming. Programs and data were recorded and entered into computers via 80 column punch cards. Each of the 80 columns could contain exactly one byte of information, which corresponded to one of the four digits needed to represent a year. The cumbersome nature of punched cards encouraged using as few of them as possible.

Although programmers and managers knew they had built software with latent defects in it, no one thought that software written in the 60s and 70s would survive to the Year 2000. Compounding the problem, newer software had to interface and share data with the older software. Although the new software could have handled dates internally in four digit formats and swapped data in two digit formats with the older software, to do so added complexity and hence added cost to new software. The net result was that the two-digit standard for representing years continued much longer than anyone would have guessed.

When will Y2K problems start?

Y2K problems have already surfaced in many places. Cap Gemini, a technology consulting firm, reported that as of December 1997, 7% of a group of 128 large U.S. companies had experienced Y2K related problems.[3] By March 1998, that number leaped to 37%. The Gartner Group, an information technology research company, has developed a model to predict the rate of occurrence of Y2K problems. This prediction is based on data collected quarterly from over 15,000 firms and government organizations in 87 countries. Gartner estimates a rapid increase in problems in 1999 with a peak sometime after January 1, 2000. Problem occurrences will drop off after 2000, but will still occur for another 3 to 5 years at a lower level. Finally, the Information Technology Association of America has reported that about half the major corporations in America have already experienced some form of Y2K disruption as of March 25, 1998.[4]

How can we fix Y2K and how long will it take?

It is beyond the scope of this report to cover the technical nuances of these various solutions. However, various techniques are briefly described in Appendix III.

How much are Y2K fixes going to cost?

There is no generally agreed upon answer to this question. The Gartner Group's estimate of $600 billion worldwide is a frequently cited number. Another number from a reputable source is that of Capers Jones, Software Productivity Research, Inc. of Burlington, MA. Jones' worldwide estimate is over $1.6 trillion.[5] Part of the difference is that Jones' estimate includes over $300 billion for litigation and damages but Gartner's does not. A sense of the scale of the cost can be gained from looking at the Y2K costs of six multinational financial services institutions; Citicorp, General Motors, Bank America, Credit Suisse

Group, Chase Manhattan and J.P. Morgan. These six institutions have collectively estimated their Y2K costs to be over $2.4 billion. Additionally, the estimated cost of Y2K repairs is increasing, as shown in figure 2.

Figure 2. Y2K Repair Estimates[6]

Company	*Past Est. (millions)*	*New Est. (millions)*
Aetna	$139	$195
ATT	$300	$900
Bankers Trust	$180-$230	$220-$260
Cendant	$25	$53
Chase Manhatt.	$300	$363
General Motors	$400-$500	$890
McDonald's	$8	$30
Merrill Lynch	$375	$560
Sears	$63	$143
Xerox	$116	$135

Can't we develop an easy Y2K fix?

Popular sentiment suggests that a technological quick fix will appear just in time to kill the millennium bug. So far, "quick fix" claims have proved to be claims for a particular product that may show promise in one particular application, for example, finding where the actual dates and date processing routines are hidden in a program.

Software programs and computer hardware vary too greatly to be fixed by one solution. Currently, there are over 500 programming languages in use. A universal or broadly applicable Y2K solution would have to be compatible with many or most of these languages. Additionally, finding all the dates and date processing in an estimated 36,000,000 programs[7] is an enormous task difficult to automate.

The embedded processors pose another problem. Although the percentage of embedded chips with a Y2K problem is estimated to be relatively small, potentially millions of chips exist that may have to be replaced. Unfortunately, most of them are not readily accessible or easily modified.

Where can I learn more about the Y2K problem?

Many solid references can be found in the endnotes of this section and elsewhere in this report. An enormous amount of Y2K information resides on the Internet. However, legitimate information is buried among overstated rumors and half-truths. As with most other information derived from Internet sources, Y2K information must be verified for accuracy.

Additional information can be obtained through the Committee's website at www.senate.gov/~y2k and the President's Council on Year 2000 Conversion's website at www.y2k.gov.

CRITICAL INFRASTRUCTURES

Critical infrastructures can include both computerized services and physical services essential to minimum functioning of economy and government. More than abstract systems, critical infrastructures enable the average person to use an