

We're concerned about several computer systems that are controlling sensitive services and facilities around the world. Each of these systems run their own proprietary operating systems, all of which appear to have different methods of representing dates and times. Because of this, they thankfully were not affected by the Y2K problem, but it would make us feel better if you could find out the exact dates on which we will start encountering problems so that we can prioritize resources on the ones we need to fix first.

We already did some work for you by figuring out how the systems in each facility keep track of time and have described them below. Now it's up to you to find out when they'll fail and what to do about it. We think you'll excel at this.

- **Depository for Pathogenic Agents:** The date is kept using 16 bits. The first 7 bits represent the number of years that have elapsed since the year 1950. The next 4 bits represent the current month (a value from 1 to 12). The final 5 bits represent the day (from 1 to 31). All fields are unsigned integers. This date value is updated at the end of every day.

- **Autoridad Hidroeléctrica de Centroamérica:** A 16-bit binary-coded decimal counter is used to keep track of the number of days that have elapsed since December 1, 1990 (which is day 0). Each BCD digit is 4 bits in length. This counter is incremented at the end of every day.

- **Yamato Nuclear Research Facility (ヤマト原子研究施設):** The system keeps time in a 32-bit unsigned integer that represents the number seconds that have elapsed since midnight on January 1, 1900. This value is incremented every second.

- **Système National de Télécompensation:** Time is kept using 32 bits, but the first 26 bits as an unsigned integer represent the number of minutes elapsed since midnight on January 1, 1972. The latter 6 bits as an unsigned integer represent the number of seconds that have elapsed during any particular minute represented by the first 26 bits. This value is updated every second.

We've included the relevant source code in an appendix on a separate page in case you need any clarification, but you should have enough information to figure it out based on our descriptions alone.

Note: When you think you've figured out the date of failure for any particular system, submit it to us and we can quickly verify whether or not it is the date of failure for any of the systems. Submit each date as a number in the format YYYYMMDD (e.g., for November 6, 2010, you would submit 20101106). Once you determine all the dates, you can start thinking about how to apply what you find to each facility.



Appendix: This is the relevant source code in case you need any clarification, but you should have enough information to figure it out based on our descriptions alone.

- **Depository for Pathogenic Agents:** The following C routine is run at the end of every day:

```
void NextTick(uint16_t &date)
{
    static const uint16_t daysInMonth[] =
        {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

    uint16_t year = (date >> 9) + 1900;
    uint16_t month = (date >> 5) & 0x000F;
    uint16_t day = date & 0x001F;

    date++;

    if ((date & 0x001F) == 0) date++;
    else if ((day >= daysInMonth[month - 1]) &&
        ((month != 2) || (day == 29)) ||
        (((year % 4 != 0) || (year % 100 == 0)) && (year % 400 != 0))))
    {
        date &= 0xFFFF0;
        date += 0x0021;
    }

    if (((date >> 5) & 0x000F) > 12)
    {
        date &= 0xFFE1F;
        date += 0x0220;
    }
}
```

- **Autoridad Hidroeléctrica de Centroamérica:** The following C routine is run at the end of every

day:

```
void NextTick(uint16_t &daysElapsed)
{
    daysElapsed++;
    if ((daysElapsed & 0x000F) > 0x0009) daysElapsed += 0x0006;
    if ((daysElapsed & 0x00F0) > 0x0090) daysElapsed += 0x0060;
    if ((daysElapsed & 0x0F00) > 0x0900) daysElapsed += 0x0600;
    if ((daysElapsed & 0xF000) > 0x9000) daysElapsed += 0x6000;
}
```

- **Yamato Nuclear Research Facility (ヤマト原子研究施設):** The following C routine is run every

second:

```
void NextTick(uint32_t &secondsElapsed)
{
    secondsElapsed++;
}
```

- **Système National de Télécompensation:** The following C routine is run every second:

```
void NextTick(uint32_t &timeElapsed)
{
    timeElapsed++;
    if ((timeElapsed & 0x3F) > 59) timeElapsed += 4;
}
```

