Solution for "Friday the Thirteenth" USACO Training Pages

by Arnav Joshi

This problem is a straightforward simulation problem. At any given month, to find the day the 13th falls on for the next month, you must move forward by mod 7 of the number of days in the month. For example, if in January the 13th falls on a monday, in February it would fall on Thursday because January has 31 days and 31 mod 7 is 3 so we must move forward by 3 days. By simulating this process for N years, and storing the number of occurrences of each day in an array, we can easily arrive at our answer. One thing we have to be careful of is leap years, which occur every 4 years unless the year is divisible by 100 (but they do occur if the year is divisible by 400), and add 1 day to February.

```
read n
finalYear = 1900 + n
// the first day of the first month is given as monday
firstDayOfMonth = 0
days[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31}
leapYearDays[] = {31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31}

// first, we go through each year
for i from 1900 through finalYear-1:
      // within each year, we go through each month
      for j from 0 through 11:
            // add 1 for the day that the 13th falls on
            answer [(firstDayOfMonth + 12) mod 7) += 1

            // find the first day of next month
            if isLeap(i) = true:
                  firstDayOfMonth += days[j]
            else:
                  firstDayOfMonth += leapYearDays[j]

// now we print our answer starting with saturday
print answer[5], answer[6], answer[0], answer[1], answer[2], answer[3],
answer[4]
```

In the following code, we use the `isLeap(a)` method. This method checks the year and uses the rules stated in the problem to check if the year is a leap year or not.

```
isLeap(a):
      if a % 4 = 0 and a % 100 is not 0:
            return true
      else if a % 400 = 0:
            return true
```

```
return false
```