**Test Case 2 Problem**

**2.4 Sample 4 Calendar**

1. **Description of the contest problem.**

A calendar is a system for measuring time, from hours and minutes, to months and days, and finally to years and centuries. The terms of hour, day, month, year and century are all units of time measurements of a calendar system.

According to the Gregorian calendar, which is the civil calendar in use today, years evenly divisible by 4 are leap years, with the exception of centurial years that are not evenly divisible by 400. Therefore, the years 1700, 1800, 1900 and 2100 are not leap years, but 1600, 2000, and 2400 are leap years.

Given the number of days that have elapsed since January 1, 2000 A.D, your mission is to find the date and the day of the week.

**Input**

The input consists of lines each containing a positive integer, which is the number of days that have elapsed since January 1, 2000 A.D.(Saturday) The last line contains an integer −1, which should not be processed.

You may assume that the resulting date won’t be after the year 9999.

**Output**

For each test case, output one line containing the date and the day of the week in the format of "YYYY-MM-DD DayOfWeek", where "DayOfWeek" must be one of "Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday" and "Saturday".

|  |  |
| --- | --- |
| **Sample Input** | **Sample Output** |
| 1730  1740  1750  1751  -1 | 2004-09-26 Sunday  2004-10-06 Wednesday  2004-10-16 Saturday  2004-10-17 Sunday |

**Source: ACM Shanghai 2004 Preliminary**

**IDs for Online Judge: POJ 2080, ZOJ 2420**

1. **Analysis of the problem**

Firstly, two functions are designed as follow.

1. *days\_of\_year*(*year*): Calculate the number of days in *year*. If *year* is a leap year, the number of days in *year* is 366; otherwise the number of days in *year* is 365.
2. *days\_of\_month*(*month*, *year*): Calculate the number of days in *month*, *year*. If *month* ==2 and *year* is a leap year, the number of days is 29; else the number of days is 28. If *month* *==*1, 3, 5, 7, 8, 10, or 12, the number of days is 31. If *month* *==* 4, 6, 9, or 11, the number of days is 30.

Then, we use January 1, 2000 (Saturday) as the benchmark. Suppose *year*, *month* and *day* are variables; *wstr* is a string array storing the day of the week, that is, *wstr*[0..6]={"Saturday", "Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday"}. Initially *year*=2000, *month*=1, and *day*=1. Suppose *n* is the number of days that have elapsed since January 1, 2000 A.D.. The steps finding the date and the day of the week are as follow:

Step 1 is to calculate the day of the week: Because January 1, 2000 (Saturday) is the benchmark, and *wstr*[0..6]={"Saturday", "Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday"}. Obviously, *wstr*[*n* % 7] is the day of the week.

Step 2 is to calculate *year*: While *n*≥*days\_of\_year*(*year*)), repeat statements *n*-=*days\_of\_year*(*year*); and ++*year*. When the loop ends, *year* is calculated, and *n* is the number of days in *year*.

Step 3 is to calculate *month* and *day*: While *n*≥*days\_of\_month*(*month*, *year*), repeat statements *n*-= *days\_of\_month*(*month*, *year*); ++*month*. When the loop ends, *month* is calculated. And statement *day* += *n* is to calculate *day*.

1. **The data structure and the algorithm to be used.**

**data structure: array**

**algorithm: date, linear list**

1. **Explanation of the data structure to be used and the algorithm to be used.**

Date is represented by Year, Month and Day. Problems for date type can make use of arrays as data structure. Normally there are two kinds of storage modes:

1. A linear list (array) whose data element is a structure containing year, month and day;
2. Three integer arrays which record years, months, and days respectively.

Dates are stored in a linear list. As a linear list, it is finite (the number of date elements is finite), ordered (date elements are listed one by one in a permutation) and uniformity ( the type of all date elements is same). Elements can be directly accessed. Therefore the linear list for date element is a typical linear list accessed directly.

The calculation of date and the conversion of calendar are based on a linear list. Months and days are generally represented by English words. Therefore months and days are stored in arrays of strings, and the indexes also correspond to months and days.