

Weekdays

Introduction

We all use electronic calendars almost every day and we take it for granted that the calendar knows the weekday for any given date. It is so common place that we don't spend much time thinking about how it is done. This exercise will not only show you how it is done but also that it is not as straight forward as one might think. Luckily, there is a formula (for dates after Oct. 15th 1582), derived by C.F. Gauss:

$$A = d + [2.6 \cdot m - 0.2] + y + [y/4] + [c/4] - 2 \cdot c$$

where

$$W = A \bmod 7$$

with

- $[x]$ is the Gaussoperator: $[x] := x'$ the biggest integer x' such that $x' \leq x$
- d represents the day ($1 \leq d \leq 31$)
- y represents the last two digits of the year
- c represents the first two digits of the year (actually all but the last two; c stands for century)
- m represents the month (see Table)
- the number W can be used to look up the weekday in Table

Table 1: weekday lookup by W

W	weekday
0	Sun
1	Mon
2	Tue
3	Wed
4	Thu
5	Fri
6	Sat

Table 2: m lookup by month

Month	m
Mar	1
Apr	2
May	3
Jun	4
Jul	5
Aug	6
Sep	7
Oct	8
Nov	9
Dec	10
Jan	11
Feb	12

However, there is more to it: Jan and Feb always belong to the 'old' year: Jan 2004 is actually the

11th month of 2003. (This is why Sep, Oct, Nov, and Dec are called something like 'seven', 'eight', 'nine', 'ten').

Example Calculation:

Let us calculate the weekday of Jan 1st of the year 2000. January 2000 is actually month 11 of the year 1999. This leads to the following values for the variables: $d = 1, y = 99, c = 19, m = 11$. Filling in the values we get:

$$A = 1 + [2.6 \cdot 11 - 0.2] + 99 + [99/4] + [19/4] - 2 \cdot 19$$

$$A = 1 + [28.4] + 99 + [24.75] + [4.75] - 38$$

$$A = 1 + 28 + 99 + 24 + 4 - 38 = 118$$

$$W = 118 \bmod 7 = 48 \bmod 7 = 6$$

Looking up 6 in the Table gives the final result: Jan, 1st 2000 was a Saturday.

Assignment:

Write a program Weekdays that asks the user for a certain date and calculates and prints the weekday of that date. Your program must reject dates that cannot exist.

- The user enters the date in the usual format (no year or month conversion done by the user!)
- Any impossible date is rejected with the message "not valid":
 - impossible month (<1 or >12)
 - impossible day (e.g., any day>31, a day>30 for some months, a day>28 in February of a non-leap year, etc.)

Examples

```
yyyyymmdd: 20180905  
Wednesday
```

```
yyyyymmdd: 20001301  
not valid.
```

```
yyyyymmdd: 19190303  
Monday
```