CSC 150 Programming Project 2

Due October 19

Fall, 2012

Assignment Overview

You are required to write a *C++* program that displays a twelve month calendar for any selected year. Prompt the user for the year of the calendar, and print out the calendar month by month.

Problem Statement

You must calculate the calendar according to the modern international standard calendar that was introduced by Pope Gregory XIII in the year 1582. For input years earlier than 1582, calculate them as if the modern calendar were in effect.

Program Specifications

1. Ask the user to enter the year that the calendar is for.
2. The user may enter any integer greater than zero for the year.
3. Once the year is entered, you must generate a 12-month calendar for that year.
4. The format of your calendar must match the format given in the example below.

Program notes

1. The most difficult task in this assignment is to calculate what day of the week January 1 falls on. Once that is determined, the remainder of the task involves printing one month at a time, while counting days and keeping track of the day of the week. This program can be written with one large main function, but you may want to break it up into several functions to improve readability and ease development.
2. switch statements and for loops are very useful in this program. You will also need to use the setw( ) manipulator or the .width( ) member to format the columns.

Finding the starting day of week

There are many ways to find the day-of-week for any given date. We will describe two methods and you can choose whichever one you prefer.

Gaussian Method

The Gaussian method is the most direct and compact method for computing the day of week for any day according to the international standard calendar. The formula is

$$w = \left(d + \lfloor 2.6m - 0.2 \rfloor + y + \left\lfloor \frac{y}{4} \right\rfloor + \left\lfloor \frac{c}{4} \right\rfloor - 2c\right) mod\ 7,$$

where w is the day of the week (0=Sunday and 6=Saturday).

Assume that Y is one less than the desired year when the month is January or February, and is the desired year for all other months. Then y is the last two digits of Y and c is the first two digits of Y. The variable d is the day of the month (1-31), and m is the *shifted* month (March=1, February=12).

The tricky part of this algorithm is that if the numerator of the mod function is negative, then we use its absolute value as the numerator, calculate the modulus, subtract the result from 7, and then do another modulus to ensure that the result is between 0 and 6.

For example, take the date May 12, 2012. For this date, $Y = 2012$ which means that $c = 20$ and $y = 12$. May is the fifth month, so m can be calculated as $m = \big((5+9) mod 12\big) + 1 = 3$. The day, d, is given as 12, so using the formula above, the day of week is:

$$w = \left(12 + \lfloor 2.6 \times 3 - 0.2 \rfloor + 12 + \left\lfloor \frac{12}{4} \right\rfloor + \left\lfloor \frac{20}{4} \right\rfloor - 2 \times 20\right) mod\ 7$$
$$w = (12 + \lfloor 7.6 \rfloor + 12 + \lfloor 3 \rfloor + \lfloor 5 \rfloor - 40)\ mod\ 7$$
$$w = (12 + 7 + 12 + 3 + 5 - 40)\ mod\ 7$$
$$w = -1\ mod\ 7$$

Since the numerator is negative, we replace it with its absolute value, calculate the modulus, subtract the result from 7, and then do a second modulus.

$$w = \big(7 - (1\ mod\ 7)\big)\ mod\ 7 = 6$$

So May 12, 2012 was a Saturday.

Counting days

Another method for calculating the day-of-week is to start with a date for which you know the day of week, and find the number of days between the known date and the date that you need. The number of days plus the reference day-of-week modulus 7 will give the week day for your desired date. With this method, you must account for leap years.

In the modern calendar, years that are divisible by 4 are leap years, except that years divisible by 100 are not leap years unless they are also divisible by 400. That is, there are 97 leap years every four centuries.

For example, take as the reference date Saturday, January 1 0000. If we want to find the day-of-week for January 1, 2012, we can count the number of days between the two dates (don't forget to add days for leap years), which turns out to be 734,868 days. (734,868 + 6) *mod* 7 = 0, so January 1, 2012 was a Sunday. For January 1, 2013, the difference is 735,234 days, so the day of week is (735,234 + 6) mod 7 = 2. January 1, 2013 will be a Tuesday.

Putting it all together

The general algorithm for printing out a calendar is as follows:

```
1.  Find the day of week for January 1 of the given year.
2.  Current_month = 0
3.  While Current_month is less than 12:
      1. Print heading for current_month
      2. Print spaces to align the first day  in the correct column
      3. Print a number for each day in the month, inserting space and endl as needed
      4. Current_month = Current_month + 1
```

Comments and suggestions

- Do not delay, Start writing the program early. If you wait until the night before the due date, you will have a miserable night.
- Do not try and write the entire program all at one time. Work on the program in small sections.
- Debugging Hint: Make sure your program's output matches the example given below. Then carefully test other values.
- Be sure to follow the coding style guidelines that can be found on the class website at http://www.mcs.sdsmt.edu/csc150/Course%20Information/.
- Name your code file `prog.cpp`. Points will be deducted if this naming convention is not followed.
- Your program must correctly compile in Visual C++ 2010.
- Be sure your code file is readable and neat. Do not allow lines to extend past 80 characters, use appropriate white space and make sure to use a consistent and attractive indentation scheme.

Program Submission

Submit your program code (the `prog.cpp` file only) at http://www.mcs.sdsmt.edu/submit before midnight of the due date. (Your file gets time stamped, so late submissions will be noted and may be given a late penalty!) Be sure to submit to the correct lecture section!

DO YOUR OWN WORK

Example Output

```
Enter the year: 2012

                January 2012
____________________________________________
   Sun    Mon    Tue    Wed    Thu    Fri    Sat

     1      2      3      4      5      6      7
     8      9     10     11     12     13     14
    15     16     17     18     19     20     21
    22     23     24     25     26     27     28
    29     30     31

                February 2012
____________________________________________
   Sun    Mon    Tue    Wed    Thu    Fri    Sat

                          1      2      3      4
     5      6      7      8      9     10     11
    12     13     14     15     16     17     18
```

19	20	21	22	23	24	25
26	27	28	29			

March 2012

Sun	Mon	Tue	Wed	Thu	Fri	Sat
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

April 2012

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

May 2012

Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

June 2012

Sun	Mon	Tue	Wed	Thu	Fri	Sat
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

July 2012

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

August 2012

Sun	Mon	Tue	Wed	Thu	Fri	Sat
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

September 2012

Sun	Mon	Tue	Wed	Thu	Fri	Sat
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

October 2012

Sun	Mon	Tue	Wed	Thu	Fri	Sat
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

November 2012

Sun	Mon	Tue	Wed	Thu	Fri	Sat
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

December 2012

Sun	Mon	Tue	Wed	Thu	Fri	Sat
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					