Lab Assignment #5

Purpose: practice selection statements and control of logical flow.
Collaboration: You should work in a group of two.

Getting started

Change into your cse220 directory
Create a new directory called lab05
Change into the new directory
Implement the following program in your lab05 directory

Day of week

Write a program DayOfWeek.c that given a date, determines what day of the week it falls on.
Your program will ask the user to enter a date in the following format: mm/dd/yyyy
Your program then computes the day of that date (Monday, Tuesday, etc.) and informs the user by writing out the following statement:

This falls on a day-of-week: xxxxx

You can compute the day of the week for a given date as follows:

(1) Compute the month coefficient (let’s call it mc) for each month according to the following mc-table:

\[
\begin{array}{c|c|c}
\text{Month} & \text{mc} & \text{Month} & \text{mc} \\
\hline
\text{Jan} & 0 & \text{Jul} & 5 \\
\text{Feb} & 3 & \text{Aug} & 1 \\
\text{Mar} & 2 & \text{Sep} & 4 \\
\text{Apr} & 5 & \text{Oct} & 6 \\
\text{May} & 0 & \text{Nov} & 2 \\
\text{Jun} & 3 & \text{Dec} & 4 \\
\end{array}
\]

(2) Decrement the year by 1 if the month is Jan or Feb, otherwise keep it the same

(3) Compute the rank as follows:

(i) Compute the following value:

\[
\text{year} + \text{year}/4 - \text{year}/100 + \text{year}/400 + \text{month coefficient} + \text{day}
\]
(ii) Find the remainder of the previous value when divided by 7. You will get an integer between 0 and 6.

(iii) Find the day according to the schedule given in the following rank-table:

rank-table:
<table>
<thead>
<tr>
<th>r</th>
<th>Day</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Sunday</td>
</tr>
<tr>
<td>1</td>
<td>Monday</td>
</tr>
<tr>
<td>2</td>
<td>Tuesday</td>
</tr>
<tr>
<td>3</td>
<td>Wednesday</td>
</tr>
<tr>
<td>4</td>
<td>Thursday</td>
</tr>
<tr>
<td>5</td>
<td>Friday</td>
</tr>
<tr>
<td>6</td>
<td>Saturday</td>
</tr>
</tbody>
</table>

Compile your program and run it. Test it on a number of dates (e.g., today's date, yesterday's date, this coming weekends' dates). Verify using your calendar that the output is correct. Show your TA the test results.

Further modify your program to make sure the given date is valid before outputting the statements:
(a) Check that month is between 1 and 12
(b) Check that day is between 1 and 31
(c) (Extra exercise: you don't have to do this one if time is not enough) Check that day and month combination is valid: for example, 6/31/2013, 2/30/2015 are not valid.

If the date is not valid, inform the user that the entered date is not valid and abort the program, otherwise continue the program execution. Compile you program and run it. Test it on valid and invalid dates. Show your TA the test results.

Hints:
(a) Input the date into three integer variables: m, d, y
(b) Use switch statement to implement the mc-table, e.g.,

```java
switch(m) {
  case 1: case 5:
    mc = 0;
    break;
  case 2: case 6:
    mc = 3;
    break;
  ... ...
}
```
(c) Use `return 1;` to abort your program (i.e., exit the `main` function). For example:

```c
int main() {
    ... ...
    /* Abort the program if input is not valid */
    if (input_not_valid) {
        return 1;
    }
    /* Continue to process if input is valid */
    ... ...
}
```

Note that, once `return 1;` is executed, the `main` function directly exits without executing any code that is below the `return 1;` statement.