Lab #4: C Expressions & Control Flow

Getting started

Download lab materials from D2L (including this instruction and two starter codes)
Enter MimirIDE
Change into the cse220 directory
Create a new directory called lab04
Change into the new directory
Upload starter codes to MimirIDE, save them in /home/(your_username)/cse220/lab03/
Implement the programs below in your lab04 directory

Program 1 Description

Write a program that asks the user to enter their birthdate in **mm/dd/yyyy** format, and outputs the day of the week (e.g. Monday, Tuesday, Wednesday…) that the user was born on.
Call your program **FindTheDay.c**

Your program should first check if the date is valid:

The month is between 1 and 12.
The day is between 1 and 31.
The day does not exceed the number of days of the month (e.g Jan 31 is valid, April 31 is not).
You do not have to check if the year is a leap year. Just take 28 days for February.

If the date is not valid, your program should output an error message and exit.
If the date is valid, your program should output in the following format:

You were born on a (day-of-week)

(day-of-week) – Insert Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, or Sunday here.

The algorithm to compute the day-of week works as follows:

1. Compute the month coefficient, mc, for each month according to the following table:

Month	mc	Month	mc
Jan	0	Jul	5
Feb	3	Aug	1
Mar	2	Sep	4
Apr	5	Oct	6
May	0	Nov	2
Jun	3	Dec	4

2. Decrement the year by 1 if the month is Jan or Feb, otherwise keep it the same
3. Compute the rank as follows:
    a. Compute the following value: year + year/4 - year/100 + year/400 + mc + day
    b. Find the remainder of the previous value when divided by 7. You will get an integer between 0 and 6.

c. Find the day according to the following schedule:

Remainder	Day
0	Sunday
1	Monday
2	Tuesday
3	Wednesday
4	Thursday
5	Friday
6	Saturday

Compile your program and generate an executable file called **findTheDay**
After running **findTheDay,** your terminal should be similar like the following:

```
user@mimir: ~/cse220/lab04 > ./findTheDay
Enter your birthdate in mm/dd/yyyy format:
01/01/1994
You were born on a Saturday
user@mimir: ~/cse220/lab04 > ./findTheDay
Enter your birthdate in mm/dd/yyyy format:
13/01/2020
Error Message: Invalid Month
user@mimir: ~/cse220/lab04 > ./findTheDay
Enter your birthdate in mm/dd/yyyy format:
01/41/2018
Error Message: Invalid Day
```

Program 2 Description

The next program will require you to implement a program that can follow the Collatz Sequence (see here for more information: https://en.wikipedia.org/wiki/Collatz_conjecture), and tell how many steps the program takes before reaching 1.
Call your program **collatz.c**
That is, the program needs to take in an integer value, and run the following process using that integer. If the integer n is even, divide it by 2 to get n / 2. If n is odd, multiply it by 3 and add 1 to obtain 3n + 1. The new value that you generate (n/2 or 3n+1) is used to repeat the process until the result of the process is 1. You should print out each of the n/2 or 3n+1 values that are calculated as intermediate steps. You should keep track of how many steps the process takes, and output that value at the end.
Example input/output:

```
user@mimir: ~/cse220/lab04 > ./collatz
Give me a number to start:
5
5 is odd.
Next number is 16.
16 is even.
Next number is 8.
8 is even.
Next number is 4.
4 is even.
Next number is 2.
2 is even.
Next number is 1.
That took 5 steps.
```