CSCI 2170 OLA 1

Write a C++ program that, given a person's birth date, computes and displays the day of the week the person was born. The description of the assignment is given below.

Check for User inputs

Your program should first check for user inputs. If the date entered is not a valid, send out an error message as shown in the example below:

1 37 2001 => Incorrect Data

If the date entered is valid, proceed to compute for the corresponding week of the day.

Determine the day of the week one was born

We use the formula called: **Zeller's Rule** to compute the day of the week for a person's birthday. The following formula is named Zeller's Rule after a Reverend Zeller. Here's the formula:

f = d + (13*m-1)/5 + D + D/4 + C/4 - 2*C. ← declare all variables to be of integer type

- **d** is the day of the month. Let's use January 29, 2064 as an example. For this date, d = 29.
- **m** is the month number. Months have to be counted in a special way for Zeller's Rule: March is 1, April is 2, and so on to January is 11, and February is 12. (This makes the formula simpler, because on leap years February 29 is counted as the last day of the year.) Because of this rule, January and February are always counted as the 11th and 12th months of the previous year. In our example, m = 11. (use multi-way if statement to compute the m value based on month)
- If the original birth month is January or February: subtract 1 from the year value. In our case, birth month is January, therefore, **modified year** = year – 1 = 2064 - 1 = 2063.
  For all other birthday months, do not subtract 1 from the birth year, i.e., modified year = year.
- **D** is the last two digits of the modified year. In this example, D=modified year % 100 = 2063%100=63**.**
- **C** stands for century: it's the first two digits of the modified year value. In our case, C = 2063/100= 20.

Now let's substitute our example numbers into the formula:

f = d + [(13*m-1)/5] + D + [D/4] + [C/4] - 2*C
= 29 + [(13*11-1)/5] + 63 + [63/4] + [20/4] - 2*20
= 29 + [28] + 63 + [15] + [5] – 40
= 100**.**

This f number modulo 7 gives the day of week: (0:Sunday, 1:Monday, 2:Tuesday, 3:Wednesday, 4:Thursday, 5:Friday, 6:Saturday). Use multi-way if statement to display the day of the week. For example, 100%7= 2. Therefore Jan 29th, 2064 is on Tuesday. ***If the remainder is negative, add 7 to the remainder.***

Given an example data as

6 15 1988
1 15 2001
1 37 2001
5 12 2005
2 19 2000
2 29 2001

The program should output the following (the format of the output should be exactly as shown below):

6 15 1988 => Wednesday
1 15 2001 => Monday
1 37 2001 => Incorrect Data
5 12 2005 => Thursday
2 19 2000 => Saturday
2 29 2001 => Incorrect Data

Documentation and program indentation and formatting

- Make sure to write the program with indentation and documentation style as discussed in "General Program Requirements for Open Labs" on the course web page: https://www.cs.mtsu.edu/~cen/2170/private/ola/programrequirements.pdf.

A program template is provided for you to start the project. You may download this program start template from the course page.

```
/*
Assignment: OLA1
Class: CSCI 2170-00X
Course Instructor: Dr. Li
Due Date:  put in due date

Description: This program determines the day of the week
a person is born, using Zeller's formula. The program reads person's
birth date (month, day, year) from a datafile and outputs the day of the week a person is
born. If birth date is invalid, the output is an error message.

Given the example data file:
6 15 1988
1 15 2001
1 37 2001
5 12 2005
2 19 2000
2 29 2001
4 31 2009
1 27 2021
2 29 2021

The program output should be:
Wednesday
Monday
Incorrect Data
Thursday
Saturday
Incorrect Data
Incorrect Data
Wednesday
Incorrect Data
*/
```

```
#include <iostream>  // for cin, cout, endl, etc
#include <fstream> // for ifstream
#include <cassert>  // for assert
using namespace std;

int main()
{
	int day; // a person's day of birth from user input.
	int month; // a person's month of birth from user input.
	int year; // a person's year of birth from user input.
	// declare other variables here

	ifstream myIn("date.dat");
	assert(myIn);

	// This loop will read one birthday at a time til the end
	// of the data file is reached
	// For each birthday read:
	// (1) check whether the birthday is valid
	//     if it is not valid, display the message
	// (2) if it is valid compute and display the day of the week
	while (myIn >> month >> day >> year)
	{

	   cout << month << " " <<  day << " " << year << " => ";


	   // Check for birthday validity


	   // if the birthday is valid, compute and display its corresponding weekday



	}// end while

	myIn.close();
	return 0;
}
```