COMP 1601 Computer Programming I

Assignment # 4

Date Due: **Wednesday November 29th, 2017 at 11:00pm**

Instructions:

- Create a folder. The name of the folder is your UWI student Id number.
- Create and test the program, `Temperature.cpp` as detailed below.
- Please include your name and UWI student Id and a *statement indicating to what extent you think your program is working correctly* as comments at the top of the program.
- The *.cpp file, the input file used and the output file generated by your program* must be saved in the folder created above.
- *Ensure* that the *correct* files are in the folder. Submission of incorrect files will be heavily penalized.
- Zip the folder and upload the zipped folder to myeLearning no later than **29th November, 2017 at 11:00pm.**
- Absolutely *no late* or *emailed* submissions will be accepted.

A file, *temp.txt*, contains the daily minimum and maximum temperatures for a given year. The first line of data in the file contains the year. After this line, each line of data contains two values, the minimum and maximum temperatures for each day of the year (in sequential order). The temperatures are in degrees Celsius (^oC).

This assignment requires you to write a program, `Temperature.cpp`, which uses arrays to store the minimum and maximum temperature for each day of the given year. Various analyses are then performed on the data stored in the arrays. Your program must provide the following functionality **in the order given**:

a) Write a function *toFahrenheit*, which accepts a temperature in degrees Celsius (^oC) and returns the equivalent temperature in degrees Fahrenheit (^oF) using the formula,

$$^oF = (^oC * 9/5) + 32$$

b) Write a function *isLeapYear* which accepts a *year* as a parameter (an integer value) and returns 1 (or *true*) if *year* is a leap year and 0 (or *false*) otherwise. A leap year is one that is either:

- Divisible by 400
- Divisible by 4 but not by 100

For example, 2000, 2004, and 2016 were leap years but 2015, 1900, and 1800 were not.

c) Using the *isLeapYear* function, write a function *startIndex* which accepts a *year* and a *month* as parameters (integer values) and returns the starting location for that month in the arrays which store the temperature values. Note that the first day of the year is stored in location 0 and that February could have either 28 or 29 days. Examples of calling *startIndex* are given below:

```
location = startIndex (2017, 3);    // location has the value 59 (March 2017)
location = startIndex (2016, 3);    // location has the value 60 (March 2016)
location = startIndex (2016, 12);   // location has the value 335 (December 2016)
```

Your *main()* function must do the following:

d) *Read* all the data in the *temp.txt* file and store the minimum temperature in one array and the maximum temperature in another array. The temperature should be stored as Fahrenheit values. *While* reading the file, your program should find and display the highest and lowest temperatures for the given year.

e) *After* reading the file, your program should find and display:
   i. Two days in March when the temperature (in Fahrenheit) was the highest.
   ii. The day in December which had the lowest temperature (in Fahrenheit).
   iii. The day in September when the difference between the maximum and minimum temperature (in Fahrenheit) was the highest for that month.
   iv. The average minimum temperature and the average maximum temperature (in Fahrenheit) for the *month* of February; also, the days when the maximum temperature went above the average.

f) Find and display all the occasions during the year (if any) when there were three consecutive days where the temperature went above a certain number *n*, where *n* is input by the user at the keyboard.

Programming Guidelines

1) Two arrays are to be used to store the temperature data from the file, one for the minimum temperature and the other for the maximum temperature.

2) The array and its size must be declared within the *main()* function. There is no need for the arrays to be accessed within another function or to be passed as parameters to a function.

3) The amount of elements stored in the arrays will vary depending on the year. If the year is a leap year, the arrays will have 366 elements. Otherwise, they will have 365 elements.

4) You may write other functions.

5) The data file for the assignment is *temp.txt*. It is available for download at the course Web site in myElearning.

6) All output is to be written to the file results.txt.

7) Please note: location 0 in the array is day 1 in January, so location 31 will be the first day in February and so on.