LAB EXERCISE 4 (SECJ1013)

PROGRAMMING TECHNIQUE 1

SEM 1, 2024/2025

INSTRUCTIONS TO THE STUDENTS

- *This exercise must be done **individually**.*
- *Your program must follow the input and output as required in the text and shown in the examples. You must test the programs with (but not limited to) all the input given in the examples.*
- *Any form of plagiarism is **NOT ALLOWED**. Students who copied other students' programs will get **ZERO** marks (both parties, students who copied, and students that share their work).*
- *Please insert your **name, matrics number, and date** as a comment in your solution.*

SUBMISSION PROCEDURE

- *Two files are required for the submission which is the source code (the file with the extension .cpp), and the input file (input_LE4.txt).*
- *Archive both files in one .zip file and name the .zip file as 'LE4'.*
- *Submit the lab exercise .zip file via the UTM's e-learning system.*

QUESTION

Given the formula for converting Fahrenheit (F) to Celcius (C):

```
C = 5/9 x (F - 32)
```

where `C` is the unit of temperature in Celcius and `F` is the unit of temperature in Fahrenheit. Write a complete C++ program that reads in a list of data `F` from a text file (**input_LE4.txt**), then calculates the values of `C` usingthe formula given. The program should use an array to store the values of `F` as example shown in **Figure1**.

```
130.89
67.75
200.67
56.78
84.15
92.48
75.32
61.75
```

Figure 1: Example of data `F` in the input file (input_LE4.txt)

The program then prints a `summary output` onto the screen and the `detail output` into a text file as shown in **Figures 2 and 3**. Grades 'H' mean high temperature; 'M' is medium temperature and 'L' is low temperature.

```
Average of the temperature in Celcius: 35.7
Number of high temperature: 2
Number of medium temperature: 3
Number of low temperature: 3
```

Figure 2: Example of output on the screen

```
C(Celcius)          F(Farenheit)          Description
==========          ============          ===========
   54.94               130.89                  H
   19.86                67.75                  L
   93.71               200.67                  H
   13.77                56.78                  L
   28.97                84.15                  M
   33.60                92.48                  M
   24.07                75.32                  M
   16.53                61.75                  L
```

Figure 3: Example of content in the output file

Your program must define several functions at least as listed in **Table 1**. You are also required to apply the concept of parameter passing to these functions.

Table 1

Function	**Description**
`readFile`	This function reads in a list of numbers from a text file and stores them into a one-dimensional array. It receives the following parameters: a. The name of the text file to be read from b. An array to store the list of numbers read c. A variable to store the number of data read
`computeC`	This function computes the values of `C`. It receives the following parameters: a. An array that contains data `F` b. An array to store the calculated values of `C` c. The number of data
`average`	This function computes the average of a list of numbers stored in an array.
`grade`	This function determines either temperature (`C`) is high or medium or low. This function will return 'H' if **C ≥ 35 ,** 'M' if **C < 35 and C ≥ 20**, and 'L' if **C < 20.**
`writeFile`	This function prints the output file as in Figure 7. It receives the following parameters: a. An array that contains data `F` b. An array that contains data `C` c. The number of data

For printing summary output onto the screen, you may define another function or you may just put the code into the `main` function **.**