ASSIGNMENT 3

PROGRAMMING TECHNIQUE 1

SEM 1, 2020/2021

INSTRUCTIONS TO THE STUDENTS

- *This assignment must be done **in pairs** (a group consisting of 2 members).*
- *Please refer to the group list to find out your group members.*
- *Your programs 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 plagiarisms is **NOT ALLOWED**. Students who copied other student's programs/assignments will get **ZERO** mark (both parties, students who copied and students that share their work).*
- *Please insert your **name and partner's name, matrics number, and date** as a comment in your program.*

SUBMISSION PROCEDURE

- *Please submit this assignment no later than **January 24, 2021, Sunday (00:00 MYT)**.*
- *Only one submission per pairs (group) that includes two files are required for the submission which is the source codes (the file with the extension .cpp).*
- *Submit the assignment via the UTM's e-learning system.*

QUESTION 1

A telematch event has been held in Sekolah Rendah Tebing Tinggi. Three teams are allowed to participate in this match, with each team consisting of four participants. Five (5) events were contested, namely E1, E2, E3, E4, and E5. Table 1 shows the scores that have been collected by each team for the five events. Write a C++ program that can assist the telematch committee to determine the winner for these events. Your program should be able to do the following tasks:

a) The program will read input data: team ID, participant ID, and **scores** for the five events namely E1, E2, E3, E4, and E5 from an input file named "`input2.txt`" into an array `marks[12][7]` of type `int`. An example of the series of input data in the input file is shown in Figure 1.

b) The program must be able to notify the user if the input file cannot be opened (failed to open) with the proper prompt. The example for user notification where the file fails to open is shown in Figure 2.

c) Calculate the total score for each participant.

Table 1: Collected scores

Team ID	Participant ID	E1	E2	E3	E4	E5
1	1001	10	5	8	10	6
	1002	8	7	10	7	9
	1003	7	10	10	6	10
	1004	10	10	8	7	7

2	2001	7	8	10	9	10
	2002	10	8	7	8	10
	2003	8	6	8	8	10
	2004	7	8	8	8	8
3	3001	10	9	10	10	10
	3002	8	7	8	8	8
	3003	7	8	9	10	6
	3004	8	6	8	7	7

```
1 1001 10 5 8 10 6
1 1002 8 7 10 7 9
1 1003 7 10 10 6 10
1 1004 10 10 8 7 7
2 2001 7 8 10 9 10
2 2002 10 8 7 8 10
2 2003 8 6 8 8 10
2 2004 7 8 8 8 8
3 3001 10 9 10 10 10
3 3002 8 7 8 8 8
3 3003 7 8 9 10 6
3 3004 8 6 8 7
```

Figure 1: Input file named "**`input2.txt`**"

```
Sorry, input file does not exist!
Press any key to continue . . .
```

Figure 2: Example user notification in case file fails to open

d) Calculate the total score for each team.

e) Besides the function **`main()`**, the program needs to define **three** (3) other functions as described in Table 2. Use appropriate argument (if necessary) for each function.

Table 2: Description for functions

Function	Description
`displayLine()`	To display lines using the 52 characters of '-'. The function should use loop to display the line.
`findIndWinner()`	To determine the winner for individual category (selected based on the highest total score that was collected by participants). The function should accept the array for a total score of each participant as one of its argument.
`findTeamWinner()`	To determine the winner of group category (selected based on the highest total score that was collected by teams). The function should accept the array for a total score for each team as one of its argument.

f) The program needs to print out the following information. Figure 3 shows the example, run of the successful program.

- The team ID.
- The participant ID.
- The scores for the five events, E1, E2, E3, E4 and E5 for each participant.
- The total score for each participant.
- The total score for each team.
- The winner for individual category (selected based on highest total score that collected by the participants).
- The winner for group category (selected based on highest total score that collected by the teams).

```
--------------------------------------------------
Id      E1      E2      E3      E4      E5      Total
--------------------------------------------------
TEAM 1
1001    10      5       8       10      6       39
1002    8       7       10      7       9       41
1003    7       10      10      6       10      43
1004    10      10      8       7       7       42
TOTAL                                           165
--------------------------------------------------
TEAM 2
2001    7       8       10      9       10      44
2002    10      8       7       8       10      43
2003    8       6       8       8       10      40
2004    7       8       8       8       8       39
TOTAL                                           166
--------------------------------------------------
TEAM 3
3001    10      9       10      10      10      49
3002    8       7       8       8       8       39
3003    7       8       9       10      6       40
3004    8       6       8       7       7       36
TOTAL                                           164
--------------------------------------------------

Winner for Individual Category: 3001 (Team 3)
Winner for Group Category: Team 2 (Score = 166)

Press any key to continue . . .
```

Figure 3: Output of the program

QUESTION 2

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, then calculates the values of `C` using the formula given. The program should use an array to store the values of `F` as example shown in Figure 4.

```
13.29
29.76
14.81
23.78
29.37
..
```

Figure 4: Example of data `F` in the input file

The program then prints a `summary output` onto the screen and the `detail output` into a text file as shown in Figures 5 and 6. Grades 'H' mean high temperature; 'M' is medium temperature and 'L' is low temperature.

Your program must define several functions at least as listed in Table 3. You are also required to apply the concept of parameter passing to these functions.

```
Average of the temperature: 32.3
Number of high temperature: 2
Number of medium temperature: 20
Number of low temperature: 8
```

Figure 5: Example of output on the screen

```
C(Celcius)          F (Farenheit)      Description
   ==========    =========                =====
   54.94               130.89                 H
   19.86                67.75                 L
   93.70               200.67                 H
   13.77                56.78                 L
    :                     :                    :
    :                     :                    :
```

Figure 6: Example of content in the output file

Table 3

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 **.**