ASSIGNMENT 2
PROGRAMMING TECHNIQUE I (SECJ/SCSJ 1013)

INSTRUCTIONS TO THE STUDENTS:

- *This assignment must be done **in pairs** (you can choose your own partner).*
- *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 program/ assignment will get **ZERO** mark (both parties, student who copied and student that share their work).*
- *Please insert your **name, IC Number, section of your class and date** as a comment in your program.*

SUBMISSION PROCEDURE:

- *Please submit this assignment no later than **November 30, 2019, Saturday (3.00 pm)***
- *Only one submission per pairs (partners) that includes 4 files are required for the submission which is the source code (the file with the extension .cpp) and the input file (the file with the extension .txt).*
- *Submit the assignment via the UTM's e-learning system.*

QUESTION 1

Ministry of Higher Education, Malaysia is required to prepare a report of the number of students' intake, enrolment and output in public universities (2015). Write a complete C++ program to calculate the total and average number of students' intake, enrolment and output in public universities for 2015. Then, find the highest and lowest number of students' intake, enrolment and output. Finally, find the range of the number of students' intake, enrolment and output. Write the program according to the following tasks:

Task 1: Write the definition of function **`getInput`**. The function must read inputs from an input file named "**`input1.txt`**" consisting of the list of public universities in Malaysia along with its number of students' intake, enrolment and output. The read data are then stored in arrays accordingly. **Figure 1** shows the input file of the program. ***Input Validation:*** You should ensure that the program will only continue reading the input file if it is successfully opened, otherwise print the error message and terminate the program.

Task 2: Write the definition of function **`calTotal`**. This function calculates the sum of elements of an array.

Task 3: Write the definition of function **`getLowest`**. The function finds the index with the lowest value in the array.

Task 4: Write the definition of function **`getHighest`**. The function finds the index with the highest value in the array.

Task 5: Using appropriate functions, read inputs from the input file and print it into the output file named "**`output.txt`**". ***Note:*** Use proper output formatting.

Task 6: Using appropriate functions, calculate the total and average of students' intake, enrolment and output. Then, print it into the output file. ***Note:*** Use proper output formatting.

Task 7: Using appropriate functions, find the highest and lowest number of students' intake, enrolment and output. Then, print it along with the name of university into the output file. Finally, find the range of the number of students' intake, enrolment and output. Then, print it into the output file.

Task 8: You should ensure the program is able to run and display correct output in the output file.

Figure 2 shows the output file of the program.

```
UM    8093 27452 6328
USM   7718 30853 6743
UKM   8109 27239 4765
UPM   8706 30670 7082
UTM   7328 31066 6997
UUM   7254 29143 6709
UIAM 10366 31526 5460
UniMAS 5578 16962 4579
UMS   5041 18531 4064
UPSI 5665 21587 11807
UiTM 65207 174755 38576
UniSZA 3523 9947 2400
UMT 3346 10665 2317
USIM 3675 14781   893
UTHM 4847 16436   4362
UTeM 3148 12370   2428
UMP   2838 9909 2122
```

```
UniMAP 4053 13769 2452
UMK   2291 9882 1062
UPNM 1341 3095 1308
```

Figure 1: Input file "**`input1.txt`**"

```
        NUMBER OF STUDENTS' INTAKE, ENROLMENT AND OUTPUT
                IN PUBLIC UNIVERSITIES (2015)

-----------------------------------------------------------------
  UNIVERSITY            INTAKE        ENROLMENT       OUTPUT
-----------------------------------------------------------------
     UM                   8093          27452           6328
     USM                  7718          30853           6743
     UKM                  8109          27239           4765
     UPM                  8706          30670           7082
     UTM                  7328          31066           6997
     UUM                  7254          29143           6709
     UIAM                10366          31526           5460
     UniMAS               5578          16962           4579
     UMS                  5041          18531           4064
     UPSI                 5665          21587          11807
     UiTM                65207         174755          38576
     UniSZA               3523           9947           2400
     UMT                  3346          10665           2317
     USIM                 3675          14781            893
     UTHM                 4847          16436           4362
     UTeM                 3148          12370           2428
     UMP                  2838           9909           2122
     UniMAP               4053          13769           2452
     UMK                  2291           9882           1062
     UPNM                 1341           3095           1308
-----------------------------------------------------------------
     TOTAL              168127         540638         122454
     AVERAGE           8406.35       27031.90        6122.70
-----------------------------------------------------------------

THE LOWEST NUMBER OF STUDENTS' INTAKE    = 1341 (UPNM)
THE LOWEST NUMBER OF STUDENTS' ENROLMENT = 3095 (UPNM)
THE LOWEST NUMBER OF STUDENTS' OUTPUT    = 893 (USIM)

THE HIGHEST NUMBER OF STUDENTS' INTAKE    = 65207 (UiTM)
THE HIGHEST NUMBER OF STUDENTS' ENROLMENT = 174755 (UiTM)
THE HIGHEST NUMBER OF STUDENTS' OUTPUT    = 38576 (UiTM)

THE RANGE OF NUMBER OF STUDENTS' INTAKE    = 63866
THE RANGE OF NUMBER OF STUDENTS' ENROLMENT = 171660
THE RANGE OF NUMBER OF STUDENTS' OUTPUT    = 37683

-----------------------------------------------------------------
```

Figure 2: Output file "**`output.txt`**"

QUESTION 2

A tele match 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 which can assist the tele match 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`**. Example of the series of input data in input file is shown in Figure 3.

b. The program must be able to notify the user if the input file cannot be opened (failed to open) with proper prompt. The example for user notification where the file fails to open is shown in Figure 4.

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 3: Input file named "**`input2.txt`**"

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

Figure 4: 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 4 shows the example, run of the successful program.

i. The team ID.

ii. The participant ID.

iii. The scores for the five events, E1, E2, E3, E4 and E5 for each participant.

iv. The total score for each participant.

v. The total score for each team.

vi. The winner for individual category (selected based on highest total score that collected by the participants).

vii. 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 4: Output of the program