PROJECT PROGRAMMING TECHNIQUE 1

SECTION 02, SEM 1, 2023/2024

INSTRUCTIONS TO THE STUDENTS

- *This case study must be done a group consisting of 3/4 members.*
- *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.*
- *Choose only **2 case study** out of provided 3 case studies.*
- *Any form of plagiarisms 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 and member's name, matrics number, and date** as a comment in your program.*

SUBMISSION PROCEDURE

- *Please submit this assignment no later than **January 21, 2023, Sunday.***
- *Only one submission per pair (group-LEADER) that includes one file is required for the submission which is the source code (the file with the extension .cpp).*
- *Submit the assignment via the UTM's e-learning system.*

CASE STUDY 1

The % Daily Value (or %DV) shows **how much of a nutrient is in one serving** of a packaged food. The %DVs are based on the values for key nutrients, which are the amounts (in grams) of nutrients recommended per day for individuals 4 years of age and older. To calculate the %DV of a packaged food, you have to divide the nutrient amount per serving by the daily recommended value and multiply by 100. To know whether you are getting enough amount of a certain nutrient from a packaged food, the following level of categorization applies:

- 5% or less of a nutrient per serving is low
- More than 5% but less than 20% of a nutrient per serving is moderate
- 20% or more of a nutrient per serving is high

Write a complete C++ program that reads a list of data (cereal type, carbohydrate in ounce, protein in ounce, fat in ounce) from an input file. Prepare the input file as shown in Figure 1.1 with data for 10 types of cereal. For the first four types of cereal (Cereal A, B, C, D), use the values as shown in the figure below. Complete the file for the remaining six types of cereal with your own values.

Cereal A	2.64	0.26	0.36
Cereal B	0.34	0.53	0.07
Cereal C	2.58	0.28	0.63
Cereal D	1.95	0.44	0.15

Figure 1.1: Input File

By reading the input file that you have prepared, your program should be able to:

(a) Calculate the conversion of the nutrient values in ounce (oz) into gram (g), with two decimal places. Note: 1oz = 28.35g
(b) Calculate the %DV of each nutrient (g) for each cereal. Note: The daily recommended value for all three nutrients are as follows:

Carbohydrate – 300g Protein – 50g Fat – 65g

(c) Calculate the average %DV of Carbohydrate, Protein and Fat for all cereals.
(d) Determine and display the category of each nutrient, either low, moderate or high.
(e) Output on screen: Ask the user to enter choice of nutrient (carbohydrate, protein, fat) and level (low, moderate, high) and display the food listed under that category (e.g. Figure 1.2)

```
     1. Carbohydrate
     2. Protein
     3. Fat

Please enter your choice>> 2



     1. Low
     2. Moderate
     3. High

Please enter your choice>> 3



List of cereal(s) with high amount of protein:

     1. Cereal B
     2. Cereal D


Choose again? Y/N >> N
```

Figure 1.2: Screen output

(f) Output on file: The program should produce an output file with the formatting shown in Figure 1.3. The corresponding level (low, moderate, high) for each percentage values also needs to be displayed, as below.

```
Type          Carbohydrate          Protein               Fat
========      =============         =======               ===
Cereal A      24.9%(high)           14.7%(moderate)       15.7%(moderate)
Cereal B       3.2%(low)            30.1%(high)            3.0%(low)
Cereal C      24.4%(high)           15.9%(moderate)       27.5%(high)
Cereal D      18.4%(moderate)       24.9%(high)            6.5%(moderate)


The four cereal types produce an average %DV of:
Carbohydrate : 18%(moderate)
Protein      : 21%(high)
Fat          : 13%(moderate)
```

Figure 1.3: output file

(g) Use array (one-dimension or two-dimension) to store the input data from file and the output data.

(h) The program should be written in several user-defined functions for example calconvert()function to convert ounce value to gram, calDV()function to calculate %DV, category()function to categorize each nutrient value, etc. Each function must be implemented with the concept of parameter passing. Use appropriate arguments for each function.

CASE STUDY 2

You have been hired under a student working scheme program. Your first task is to write a C program to assist in one of the professors in the faculty to grade the final exam of his/her students. The exam consists of 20 multiple-choice questions. Each question has one of four possible answers: A, B, C, or D. The program will read in the students' answers and the correct answers from files and prints out the result onto the screen and also the file output. There are at least 15 students in class.

Input

The students' answers are stored in a data file named "StudentAnswers.dat" as shown in Figure 2.1. In this file, the first column represents the students' name (1 word), second column represent the students' ID and the third column until the last column represent the students' answer of question 1 until question 20. In this example (Figure 1.1), there are only five students. You have to complete this input file with data at least for 15 students and cover all situations. The correct answers for all the questions are stored in a text file named "CorrectAnswers.txt" as shown in Figure 2.2.

Output

The program needs to determine and print out the following items **on the screen** as shown in Figure 2.3:

- Ask the user to enter student's ID
- Display the student's ID and student's name
- Compare the student's answers and the correct answers. Calculate and display the total number of question missed by the student.
- Display the list of the questions missed by the students, showing the correct answers and the incorrect answers by the student for each missed question.
- Calculate and display the percentage of questions answered correctly. This can be calculated as:

$$percentage = \frac{correctly\ answered\ questions}{total\ number\ of\ questions} \times 100$$

- Display the grade of the students based on the percentage as follows:
  - $80 \leq$ percentage ≤ 100 - grade is A
  - $70 \leq$ percentage < 80 - grade is B
  - $60 \leq$ percentage < 70 - grade is C
  - percentage < 60 – grade is F

The program needs to print out the following item onto the **file output** as shown in Figure 2.4:

- Display all students' name, students' ID, students' percentage and students' grade

- The number of student is based on the input data from file "StudentAnswers.dat".
- Use array (one-dimension or two-dimension) to store the input data from file and the output data.
- The program should be written in several user-defined functions for example `readFile()` function to read input data, `compareAnswer()` function to check the student's answers, `printMissQuestion()` to display the missed questions and the correct answer, `printReport()` to display output onto the output file etc. Each function must be implemented with the concept of parameter passing. Use appropriate arguments for each function. Do not use global variables.

```
Abdullah    A19EE0180   A B A D A C C D A B C D A D C D A B C D

LuDong      A19EE0160   A B D D A C B D D B C A A D C D A B C D

Syarifah    AC12CS678   A A D C B C D D A B C C A A C D A D C D

Sivarajah   AC12CS123   C B C B A C C D B B C C A B C D C C C D

Wendy       B19EE0167   A C C D A B C C A A C D A A A D A B C D
```

Figure 2.1: file "StudentAnswers.dat"

```
A B C D A B C D A B C D A B C D A B C D
```

Figure 2.2: file "CorrectAnswers.txt"

```
Enter the student ID: B19EE0167

EXAM RESULT
Name              : Wendy
Student ID  : B19EE0167
Number of questions missed: 5
List of the questions missed:
Question        Correct Answer      Student Answer
2                 B                       C
8                 D                       C
10                B                       A
14                B                       A
15                C                       A
Percentage: 75% , GRED : B
```

Figure 2.3: example of output on the screen

LIST OF STUDENTS AND GRADES

NAME	ID	PERCENTAGE	GRADE
Abdullah	A19EE0180	85	A
LuDong	A19EE0160	70	B
Syarifah	AC12CS678	55	F
Sivarajah	AC12CS123	65	C
Wendy	B19EE0167	75	B

Figure 2.4: example of output in the output file.

CASE STUDY 3

The Maslee Mart is a convenience store company operating in the state of Johor. The company has five stores located in several branches; Johor Bahru, Segamat, Batu Pahat, Kota Tinggi and Mersing, respectively. At the end of each year, the management of the company wants to know the performance of their company. They have decided to use a computer program to help them in analyzing the company's sales. You, as a freelance programmer have been appointed to develop the program using C language. The requirements of the program are as follow:

Input:

- The program should read in sales data from a text file named "sales2014.dat" as shown in Figure 3.1.
- The format of the input file is as follows: The first to twelfth columns indicate the sales for each month, i.e., the first column is for the sales of January, second column is for February, third column is for March, and so forth. The last column indicates store branches. Note that sales in each cell is represented in multiple of RM 1000.00

Output:

- The program should print out a report into an output file as shown in Figure 3.2.
- The report should include:
    - The grand total of sales, i.e., over all stores throughout the year.
    - The average sales per month.
    - The highest sales. Print the store, month and the sales whose the highest sales.
    - The lowest sales. Print the store, month and the sales whose the lowest sales.
    - The total sales for each month. The months should be printed with their abbreviated names, such as, "Jan", "Feb", "Mar", and so forth.
    - The total sales for each store.
    - The list of profitable stores. A store is considered profitable if it manages to achieve minimum annual sales of RM600,000.00.

- Note that all money values have to be specified as with 2 decimal points, 10 spaces in width, and right-justified.

- Use array (one-dimension or/and two-dimension) to store the input data from file and the output data.
- The program should be written in several user-defined functions for example **readFile()** to read data from file, **grandTotalSales()** to calculate the total of sales over all store throughout the year, **highestSale()** to find the highest sale, **lowestSale()** to find the lowest sales, etc. Each function must be implemented with the concept of parameter passing. Use appropriate arguments for each function. Do not use global variables.

```
94  49  96  67  82  34  91  64  15  97  98  78  Johor Bahru
71  57  17  31  63  38  77  74  61  22  27  59  Segamat
36  16  30  19  29  41  23  25  22  37  28  29  Batu Pahat
87  48  49  91  72  69  13  97  43  41  29  58  Kota Tinggi
34  32  74  57  32  80  76  40  64  48  41  68  Mersing
```

Figure 3.1: File " sales2014.dat"

```
total of sales over all stores: RM 3140000.00 Average
sales per month:              RM 261666.67

The highest sales:
------------------
Store: Johor Bahru
Month: Nov
Sales: RM   98000.00

The lowest sales:
------------------
Store: Kota Tinggi
Month: Jul
Sales: RM   13000.00


Total sales by month:
Month   Sales
-----   -----
Jan   RM  322000.00
Feb   RM  202000.00
Mar   RM  266000.00
Apr   RM  265000.00
May   RM  278000.00
Jun   RM  262000.00
Jul   RM  280000.00
Aug   RM  300000.00
Sep   RM  205000.00
Oct   RM  245000.00
Nov   RM  223000.00
Dec   RM  292000.00

Total sales by store:
Store         Total Sales
-----         -----------
Johor Bahru  RM  865000.00
Segamat      RM  597000.00
Batu Pahat   RM  335000.00
Kota Tinggi  RM  697000.00
Mersing      RM  646000.00

Profitable stores:
-----------------
Johor Bahru
Kota Tinggi
Mersing
```

Figure 3.2: The output file for the sales data in Figure 3.1