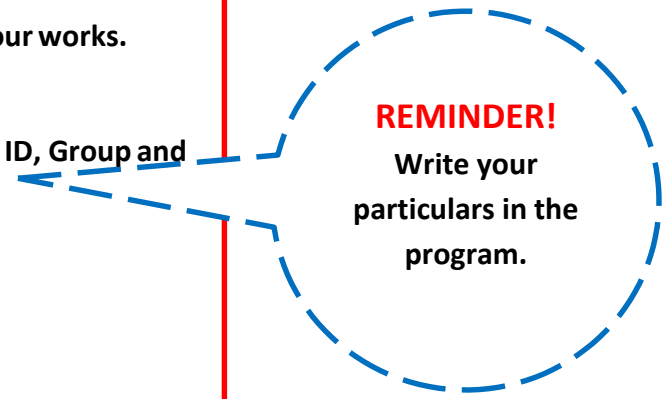
NOTE:

- Create a folder on Desktop to save your works.

- Use comment // to write your name, ID, Group and Lab Question in each program.

- Save your file as .c



LAB OBJECTIVE

At the end of this lab activity, the students should be able to:

- Read and write from/to files to solve programming problems.

QUESTION 1

The program is to calculate the summon fee for the persons in the file. Write **C code** based on the following instructions.

i. Create file variables named *input* and *output.*

ii. Open the file *summon_list.txt* for reading using *input* and *output* for append the file.

iii. Use a *while loop* statement to loop till the end of the file:

- Read all the data from the file and store them in appropriate variables.
- Based on the Table 2, determine the *summon fee* using *if else* statement.

Speed	Summon Fee
More than equal 100	300
Others	150

Table 2: Summon Fee

- Write the *name, speed, summon fee* in the output file.

iv. Close the files.

The contents of the files are shown below.

Content of *summon_list.txt* before execution

```
<Name> <speed>
Jason     95
Rameshan 120
Zafran   100
```

Content of *summon_list.txt* after execution

```
<Name> <speed> <summonfee>
Jason     95
Rameshan 120
Zafran   100
Jason 95 150.00
Rameshan 120 300.00
Zafran 100 300.00
```

QUESTION 2

Write a complete C program that creates a new file called *audio.txt* for both writing and reading. The program prompts user to enter a series of inputs to calculate the audio size. User inputs and audio size are then written into file *audio.txt*.

In *main():*

- Open a new file called audio.txt for writing and reading.
- If the file can't be opened, display error message *"File could not be opened. Program aborted*." And quit the program.
- Using a do-while loop:
  - Prompt user to enter audio channel (either 1 for mono or 2 for stereo), sampling rate, time, and bit depth.
  - Calculate audio size.
    - Formula: audio channel x sampling rate x time x (bit depth / 8)
  - Write audio channel, sampling rate, time, bit depth and audio size into file audio.txt.
  - Prompt the user whether to continue adding another audio record. Repeat the above as long as user chooses to continue.
- From the beginning of file audio.txt, read and display all data from each audio record until the end of file.
- Close file audio.txt before program ends.

SAMPLE OUTPUT

```
========================================
=        CALCULATE AUDIO DATA SIZE     =
========================================
Enter channel - [1] Mono [2] Stereo     : 2
Enter sampling rate (Hz)                : 44100
Enter time (seconds)                    : 30
Enter audio bit depth (bits)            : 16

Add another record (y/n)? y

Enter channel - [1] Mono [2] Stereo     : 1
Enter sampling rate (Hz)                : 42500
Enter time (seconds)                    : 25
Enter audio bit depth (bits)            : 8

Add another record (y/n)? n


Channel Sampling Rate   Time    Bit Depth       Audio Size
-----------------------------------------------------------------
2       44100.00        30      16              5292000.00
1       42500.00        25      8               1062500.00
```

QUESTION 3

Write the code segments based on the given descriptions:

1. Declare a file pointer called *input*. Open a file called *location.txt* for reading using this pointer.
2. If the file is not available, display *"File does not exist."*.
3. The content of the text file *location.txt* is as shown below. It includes the *location*, *latitude* and *longitude values.*

FILE *location.txt* CONTENTS

<Location> <Latitude> <Longitude>

```
UTM 1.5523763 103.63322
KLCC 3.153889 101.71333
UM 3.1185 101.665
UMS 6.0367 116.1186
UNIMAS 1.465174 110.4270601
```

Ask the user to enter a location. Check if the user's location can be found in the text file. If found, display the *location*, *latitude* and *longitude* as shown below.

SAMPLE OUTPUT

```
Enter a location: UM

Location      : UM
Latitude      : 3.1185
Longitude     : 101.6650
```

Otherwise display *"Sorry, location could not be found"*.

SAMPLE OUTPUT

```
Enter a location : MMU
Sorry, location could not be found
```