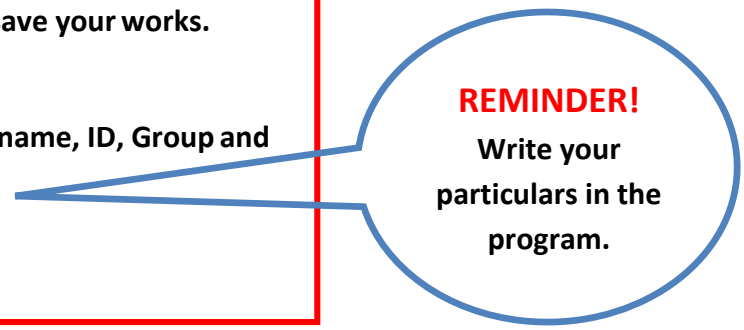
NOTE:

- 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 OBJECTIVES

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

- Declare an array (one-dimensional and two-dimensional)
- Process an array such as printing, finding minimum value or maximum value, and getting user input for the array.
- Pass array to function.
- Use array as a global variable.

QUESTION 1

Write a program to convert inches to centimeters.

- 1 inch is equivalent to **2.54** cm. Set this as constant using *preprocessor directive.*
- Declare **two (2)** arrays called ***cm*** and ***inch*** with 5 elements.
- Using *for* loop:
    - Ask the user to enter five values in inch and store them in the ***inch*** array.
    - Next, convert the value to centimeters and store them in the ***cm*** array.
- Using another *for* loop:
    - Print each value from both arrays as shown in the output below:

Sample Output

```
Enter value in inches : 1
Enter value in inches : 7
Enter value in inches : 10
Enter value in inches : 13
Enter value in inches : 20

Here are the results of the conversion.
-------------------------------------
1.00 inch(es) ===> 2.54 cm(s)
7.00 inch(es) ===> 17.78 cm(s)
10.00 inch(es) ===> 25.40 cm(s)
13.00 inch(es) ===> 33.02 cm(s)
20.00 inch(es) ===> 50.80 cm(s)
```

QUESTION 2

Write a program to identify the minimum and maximum rain fall within a 5-month duration.

- In the *main()*:
  - Declare an array named *rainfall* with the following values:
    - 5.67, 10.9, 2.03, 12.08, 7.11
  - Call function *get_min(…)* and pass the rainfall as argument.
  - Call function *get_max(…)* and pass the rainfall as argument.
  - Call function *display(…)* and pass the rainfall as argument.
  - Display the minimum rain fall data.
  - Display the maximum rain fall data.

Don't forget to write the function prototype

- In *get_min(…)*:
  - Using a *for* loop, identify the minimum rain fall value from the array and return it.
- In *get_max(…)*:
  - Using a *for* loop, identify the maximum rain fall value from the array and return it.
- In *display(…):*
  - Using a *for* loop, display the entire array content.

Sample Output

```
5 months rain fall statistics.
------------------------------
Month 1 : 5.67 ml
Month 2 : 10.90 ml
Month 3 : 2.03 ml
Month 4 : 12.08 ml
Month 5 : 7.11 ml

Minimum rain fall : 2.03 ml
Maximum rain fall : 12.08 ml
```

QUESTION 3

Write a program to convert kilogram to gram and pound.

Don't forget to write the function prototype

- Declare an array as a **global variable**, called ***kilo*** with 4 elements.

- In *main()*:
  - Call function *get_input()*.
  - Call function convert_gram().
  - Call function convert_pound().

- In *get_input()*:
  - Using *for* loop, prompt the user to input the values in kilogram and store them in the array ***kilo***.

- In *convert_gram()*:
  - Use *for* loop to convert the values in grams and display the values.
  - Formula: 1 kg = 1000 g

- In *convert_pound()*:
  - Use *for* loop to convert the values in pounds and display the values.
  - Formula: 1 kg = 2.205 pounds

Sample Output

```
Enter the values in kilogram:
3.5
6.75
0.45
2.1

Convert kilo to gram:-
3.50 kg is equals to 3500.00 grams
6.75 kg is equals to 6750.00 grams
0.45 kg is equals to 450.00 grams
2.10 kg is equals to 2100.00 grams

Convert kilo to pound:-
3.50 kg is equals to 7.72 pounds
6.75 kg is equals to 14.88 pounds
0.45 kg is equals to 0.99 pounds
2.10 kg is equals to 4.63 pounds
```

QUESTION 4

Write a program that applies two-dimensional array and calculates average for each row.

- Declare a **two-dimensional array (4 × 5)** that represents the ratings customers gave to 4 different products. Refer to the figure below for the data.
  - Row represents the product
  - Column represents the customer.

- Use **two (2)** *for* loops to calculate the average rating for each product.
  - Hint: Create a counter to count the number of customers have rated (if the value is not 0)
  - The output should be displayed as shown in the sample output.

		[0]	[1]	[2]	[3]	[4]	
Product 1 rating	[0]	2.30	4.00	5.00			Only 3 customers rated
Product 2 rating	[1]	3.00	2.10	1.00	4.00		Only 4 customers rated
Product 3 rating	[2]	1.00	2.00	3.00	4.00	5.00	5 customers rated
Product 4 rating	[3]	4.00	2.50				Only 2 customers rated

Sample Output

```
Average Rating For Each Product
-------------------------------
Product 1 : 3.77

Product 2 : 2.53

Product 3 : 3.00

Product 4 : 3.25
```