

**LAB OBJECTIVE**

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

- Use one dimensional and two dimensional arrays in a C program.
- Use built-in function **strlen()** and **strcmp()** to manipulate sequence of characters.

**PRACTICE**

1. Write the complete C program to convert inches to meter.
  - 1 inch is equivalent to 2.54 cm; set this as constant using **preprocessor directive**.
  - Declare 2 empty arrays called *cm* and *inch* with 5 elements.
  - Using a **for loop**:
    - Ask the user to enter five values in inch and store them in the *inch* array.
    - Next convert the value to meter and store them in the *cm* array.
  - Using a **do-while loop**:
    - Retrieve each value from both arrays so that the output can be displayed as shown below.

```

Enter value in inches : 1
Enter value in inches : 7
Enter value in inches : 10
Enter value in inches : 19
Enter value in inches : 23

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)
19.00 inch(es) = 48.26 cm(s)
23.00 inch(es) = 58.42 cm(s)

```

2. Write the codes that declare a two-dimensional array size 3 × 5. Using for-loops, tabulate the array with the following values and display on the output screen. [Hint: Use the **pow(...)** function]

	[0]	[1]	[2]	[3]	[4]
[0]	2	4	8	16	32
[1]	3	9	27	81	243
[2]	4	16	64	256	1024

Sample output:

2	4	8	16	32
3	9	27	81	243
4	16	64	256	1024

3. Write a program that declares a two-dimensional array ( $4 \times 5$ ) that represents the ratings customers gave to 4 different products. Use two (2) for loops to calculate the average rating for each product. The output should be displayed as shown below.

		[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:-

```
Product 1 : 3.77
Product 2 : 2.53
Product 3 : 3.00
Product 4 : 3.25
```

4. Write a program that generates build a graph based on the product rating.
- In the *main()* function:
    - Declare an integer array called *rating* that has 4 elements
    - Using for loop, ask the user to rate the 4 products (rating 1-5).
    - Using a do while loop, send each rating value to the *display\_graph(...)* function.
  - In the *display\_graph(...)* function:
    - Use a while loop to display the graph based on the value.

```
Rate product 1 : 3
Rate product 2 : 1
Rate product 3 : 4
Rate product 4 : 5

Product 1 : ***
Product 2 : *
Product 3 : ****
Product 4 : *****
```

5. You are required to calculate the bill for sending a message for a customer.
  - Declare all necessary variables.
  - Ask the user if they wish to write a message.
  - If they do:
    - Ask them to enter the message.
    - Get the length of the message using the function **strlen()**.
  - Using **if else** statement, determine the cost per character for the message based on the length.

MESSAGE LENGTH	COST
20 characters or less	RM 0.50
21 to 50 characters	RM 0.30
More than 50 characters	RM 0.10

- Calculate the bill.
- If the user wishes to stop, display the amount of messages and the total bill.

```

Enter [Y] if you want to write a message: Y

Write your message : It won't be easy but it would be worth it!

Message : It won't be easy but it would be worth it!
length  : 42
Cost    : RM 0.30
Bill    : RM 12.60

Enter [Y] if you want to write a message: Y

Write your message : Carpe Diem!

Message : Carpe Diem!
length  : 11
Cost    : RM 0.50
Bill    : RM 5.50

Enter [Y] if you want to write a message: N

Total message : 2
Total Bill    : RM 18.10

```

6. Write a program that verify password that user entered.
- In function *main()*:
    - Declare 3 string arrays to store the value of string *password*, *confirmPassword* and *status*.
    - Ask the user to enter *password* and ensure that the password length to be at least 8 characters. Otherwise ask for another *password*. If password is at least 8 characters, ask to confirm password.
    - Call function *confirmpassword(...)*, passing both *password* and *confirmPassword* as parameters.
    - Repeat 2 times using for loop.
    - Display output as shown below.
  - In function *confirmpassword (...)*:
    - Compare both values to see if they are a match (use *strcmp*).
    - Return the status whether it is a *Matched* or *Not Matched*.

Sample output:-

```

Enter your password   : DCS5038
Your password is too short!

Enter another password : DCS5038**
Re-enter password     : DCS5038*#
Password status       : Not Matched!

Enter your password   : DCS5038**
Re-enter password     : DCS5038**
Password status       : Matched!

```

7. Write a C program for a book shop.

- Declare all necessary variables.
- Ask the user whether they want to continue [enter 1].
- Ask the user to enter book code and quantity.
- Use if else statement to get the price.
- Use *strcmp()* function to determine the correct book code.

BOOK CODE	BOOK PRICE
B1001	RM 34.50
B1002	RM 77.30
B1003	RM 54.90

- Calculate total bill.
- If the user wants to stop the program, display the overall sales for all the books and the quantity sold for each book.

## SUBMISSION

Write the complete C program to identify the minimum and maximum rain fall within a 5 months duration.

- In *main()*:
  - Declare an array with the following values:  
5.67, 10.9, 2.03, 12.08, 7.11
  - Call function *get\_min(...)* and pass the array as argument.
  - Call function *get\_max(...)* and pass the array as argument.
  - Call function *get\_average(...)* and pass the array as argument.
  - Call function *display(...)*.
  - Display the minimum rain fall amount.
  - Display the maximum rain fall amount.
  - Display the average rain fall amount.
- 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 *get\_average(...)*:
  - Using a **for loop**, calculate the average value from the array and return it to *main()*.
- In *display(...)*:
  - Display the entire array content.

```
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
Average rain fall : 7.56 ml
```