Question 1

a) Write a C++ program to display the following text.

Sample Output

```
MMU Fee Structure 2022
  Programme          Duration              Fee
    Diploma           2 Years            23000
     Degree           3 Years            60000
```

b) Given below is an incomplete program. Complete the program based on the instructions given in the comments. Refer to sample program output given.

```
#include<iostream>
#include<iomanip>
using namespace std;
int main()
{
   float day1, day2, total;
   cout<<"Enter the sales for day 1 :" ;
   cin>>day1;
   //prompt user to enter the sales for second day

   //calculate the total sales;

   //display the sales figures with the total
   return 0;
}
```

Sample Output

```
Enter the sales for day 1 : 25399.75
Enter the sales for day 2 : 26536.88

QTech Sdn Bhd Sales figures for 2 days
Day 1 : 25399.75
Day 2 : 26536.88
Total : 51936.63
```

Question 2

a) Write a C++ program that calculates and displays an employee's total wages for the week. The regular hours for the work week are 40 and any hours worked over 40 are considered overtime. The employee earns RM18.50 per hour for regular hours, and RM24.35 per hour for overtime hours. The following pseudocode algorithm shows the program's logic.

Sample Output 1

```
Please enter total hours you've worked this week:65
Wages for this week are RM1348.75
```

Sample Output 2

```
Please enter total hours you've worked this week:50
Wages for this week are RM983.50
```

```
DECLARE variables as required, set regular hours = 40, set overtime pay rate=24.35, set base pay rate = 18.50
GET user input for total hours
IF total hours greater than 40
    overtime hours = total hours – regular hours
ELSE
   overtime hours = 0
   regular hours = total hours;
END IF
Regular wages =  base pay rate X regular hours
Overtime wages = overtime pay rate X overtime hours
Total wages = regular wages + overtime wages
DISPLAY the total wages
```

b) Write a complete program:

- Prompt user to enter 10 numbers.
- Determine how many inputs are odd and even and display the total of all the numbers.
- Incorporate ***for*** loop and selection structure in your solution.

Sample Output

```
Enter a number :2
Enter a number :6
Enter a number :3
Enter a number :4
Enter a number :1
Enter a number :8
Enter a number :10
Enter a number :9
Enter a number :5
Enter a number :7

There are 5 even numbers and 5 odd numbers.
The total of all the 10 numbers are 55
```

c) Write a complete C++ program based on the descriptions below:
- Get user input for a number. The program will accumulate the inputs.
- This process will be repetitive until user enter value 99 (Use a ***do-while*** loop).
- After that, the loop will terminate and the total of all the inputs will be displayed.

Sample Output

```
Enter a number :5
Enter a number :80
Enter a number :6
Enter a number :99
The total is 91
```

d) Referring to the sample output, write a program that calculates the total price of ice-cream selected by user.
- Prompt the user to enter name, ice-cream flavor and quantity.
- Use ***while*** loop to prompt the user for the correct flavor in the case that the user key in incorrectly.
- Use ***switch*** *case* statement to determine the price based on the flavor.
  - Chocolate is RM3.50, Vanilla is RM3.00, and Strawberry is RM2.50 each.
- Use ***do-while*** loop to repeat the process as long as the user wants to.
- Display the total bill.

Sample Output

```
Welcome to Yummy Ice-Cream

******Ice-Cream Flavor***********
[C] Chocolate
[V] Vanilla
[S] Strawberry

Enter name     : Megan
Enter flavor   : C
Enter quantity : 2

Do you want to buy more? <Y-Yes N-No> : Y

Enter flavor   : K
Incorrect flavor. Re-enter flavor : S
Enter quantity : 3

Do you want to buy more? <Y-Yes N-No> : N

***********RECEIPT***********
Name        : Megan
Total price : RM14.50
```

Question 3

a) This program stores employee hours worked and hourly pay rates in two parallel arrays. Observe the program and the sample output. Complete it as required based on the instruction at the comments.

```
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{
        const int NUM_EMPS = 5;
        int index;
        int hours[NUM_EMPS];
        double payRate[NUM_EMPS];
        double grossPay;

        cout << "Enter the hours worked and hourly pay rates of "
        << NUM_EMPS << " employees. \n";

        // Complete the code to store the hours worked and hourly payrate of
        each employee, then display the data.

       return 0;
}
```

Sample Output

```
Enter the hours worked and hourly pay rates of 5 employees.
Hours worked by employee #1: 10
Hourly pay rate for employee #1: RM2.40
Hours worked by employee #2: 21
Hourly pay rate for employee #2: RM3.00
Hours worked by employee #3: 12
Hourly pay rate for employee #3: RM20
Hours worked by employee #4: 13
Hourly pay rate for employee #4: RM13
Hours worked by employee #5: 19
Hourly pay rate for employee #5: RM1.00

Here is the gross pay for each employee:
Employee #1: RM  24.00
Employee #2: RM  63.00
Employee #3: RM 240.00
Employee #4: RM 169.00
Employee #5: RM  19.00
```

b) Complete the C++ program:

- Use a 2D array and initialize it with the values as the given table below.
- Use nested for loops to output the values.

Row index/ Column Index	[0]	[1]	[2]
[0]	10	20	30
[1]	40	50	60

Sample Output

```
Row 1 :  10  20  30
Row 2 :  40  50  60
```

c) Write a program to record temperature for the number of days required by the user.

- Prompt the user to input the number of days.
- Create a dynamic array to store the temperature.
- Call function *average()* to calculate the average temperature.
- Use pointer notation rather than array notation whenever possible.

Sample Output

```
How many days would you like to record the temperature? 3

Enter temperature day 1 : 32.5
Enter temperature day 2 : 37.8
Enter temperature day 3 : 36.7

Average temperature : 35.67 Celcius
```

Question 4

Write a program that dynamically allocates an array large enough to hold a user-defined number of test scores. Once all the scores are entered, the array should be passed to a function named *sort()* that sorts them in ascending order. Another function named *average()* should be called to calculate the average score. The program should display the sorted list of scores and averages with appropriate headings. Use pointer notation rather than array notation whenever possible.

Sample Output

```
How many test scores will you enter? 3
Enter test score 1: 70
Enter test score 2: 90
Enter test score 3: 80

The test scores in ascending order, and their average, are:

 Score
 -----

 70.00
 80.00
 90.00

Average score:  80.00
```