Objectives:

1) Understand how to edit, compile and execute C computer codes.
2) Understand C programming: sequential code structure - revisited.

Q1) Using C computer programming language, answer the following:

A/ Declare a variable **"grade"** as a **character** with an initial value of **A**

....

B/ Define a constant **"PI"** with the value of **3.14159**

....

C/ Write an assignment statement to perform $r = 2x + \frac{y}{2z}$

....

Q2) Show the output of the following code segments:

A/
```c
int x1=1, x2=1, i, y, z;
float f;
y = 5 + x1--;
z = 5 + ++x2;
i=35/10;
f = (float) 10/4 * i;
printf("Y=%d\n",y);
printf("Z=%d\n",z);
printf("F=%.2f\n",f);
```

B/
```c
int  i = 4 , j = 3 , k = 6 , m = 4;
i * = j + k;

printf("I=%d\n",i);
```

Q3) Draw a <u>flowchart</u> to read five numbers from keyboard and calculate and display the average of these five numbers on the screen. Write corresponding C code.

Flowchart	**C Code**

Q4) The following C code contains some errors. Find and correct these errors completely. Write your answer in the provided space.

```
#include<stdio.h>              ........................................

#define month=30;             ........................................

int main()                    ........................................

{integer nday,nmonth,rday;    ........................................

scanf("%f",nday);             ........................................

Nmonth=nday/month;            ........................................

rday=nday%month               ........................................

Printf("nmonth=%d",nmonth);   ........................................

printf("rday=%d,rday);        ........................................

return 0;                     ........................................
```

Q5) Write a program that prompts the user to enter two points **(x1, y1) and (x2, y2)** and displays their distances. The formula for computing the distance **(d)** is

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$. Note you can use **pow(a,0.5)** to compute $\sqrt{a}$. Let all variables be of type **double**.

Here is a sample run:

> Enter x1 and y1: 1.5 -3.4
> Enter x2 and y2: 4 5
> The distance of the two points is 8.764131

```
#include <stdio.h>
#include <math.h>
int main()
{// Variable declarations
.........................................
// Prompt the user to enter the input

.........................................

.........................................

.........................................

.........................................
// Perform the required operation(s)
.........................................
// Print the output on the computer monitor

.........................................
return 0;}
```

Q6) Write a C program which will convert the spherical coordinates into the Cartesian coordinates using the following equations.

x = ρ cosθ sinø
y= ρ sinθ sinø
z= ρ cosø

The main program should prompt the user to input ρ, θ, and ø from the keyboard. Then, the main should print the values of x, y and z in three lines as follows:

x=
y=
z=

Q7) write a complete C program which converts a decimal number into its equivalent binary number. The program should take the decimal number as input and print out the binary number. The program can be written to convert up to 8 bit numbers i.e the maximum of 255. **(Hint: Divide the decimal number by 2 and save the remainder)**

An example:
  157/ 2 = 78 with a remainder of **1**
   78 /2 = 39 with a remainder of **0**
   39/ 2 = 19 with a remainder of **1**
    19/ 2 = 9 with a remainder of **1**
     9 / 2 = 4 with a remainder of **1**
     4 / 2 = 2 with a remainder of **0**
     2 / 2 = 1 with a remainder of **0**
     1 / 2 = 0 with a remainder of **1**   **<--- to convert write this remainder first.**
So the decimal equivalent of **157** is **10011101**