Assignment 2 from UNIT- II

AQ 1: What are different data types in C Language?

Data Type In C

DATA TYPE	SIZE (BYTES)	RANGE	FORMAT STRING
Char	1	-128 to 127	%c
Unsignes Char	1	0 to 255	%c
Short Or Int	2	-32768 to 32767	%i or %d
Unsigned Int	2	0 to 65535	%u
Long	4	-2147483648 to 2147483647	%ld
Unsignes Long	4	0 to 4294967295	%lu
Flot	4	3.4e-38 to 3.4e+38	%f or %g
Double	8	1.7e-308 to 1.7e+308	%lf
Long Double	10	3.4e-4932 to 1.1e+4932	%lf

Integar Data Type

SHORT INTEGAR	LONG INTEGAR
Occupies 2 bytes in Memory	Occupies 4 bytes in Memory
Range -32768 to 32767	-2147483648 to 2147483647
Format string is %d or % i	Format string is %ld
For example: int a=2; Short int b=2;	For example: long b; Long int c;

AQ 2: Program to print the following series by using ‘for loop’
50 48 46.....1

```
#include<stdio.h>
 #include<conio.h>
 main()
{
        int i;
        for(i=50;i>=0;i=i-2)
        {
                printf("%d",i);
        }
        getch();
}
```

AQ 3: Write a program to print the following series by using 'while loop' 125 64 27 8 1

```
#include<stdio.h>
 #include<conio.h>
main()
{
        int i;
        i=5;
        while(i>=1)
        {
                printf("%d",i*i*i);
                i=i-1;
        }
        getch();
}
```

AQ 4: Write a program to print the following series by using 'do-while loop' 1 4 9 16 25

```
#include<stdio.h>
 #include<conio.h>
main()
{
        int  i;
        i=1;
        do
        {
                printf("%d",i*i);
                i++;
        }
        while(i<=5);
        getch();
}
```

AQ 5: Using break or continue statement, write a program to calculate the sum of maximum of 10 numbers. Calculates the sum until the user enters positive number

It is sometimes desirable to skip some statements inside the loop or terminate the loop immediately without checking the test expression. In such cases, break and continue statements are used.

break Statement

The break statement terminates the loop (for, while and do...while loop) immediately when it is encountered. The break statement is used with decision making statement such as if...else.

```
# include <stdio.h>
int main()
{
   int i;
   double number, sum = 0.0;
   for(i=1; i <= 10; ++i)
   {
      printf("Enter a n%d: ",i);
      scanf("%lf",&number);
      // If user enters negative number, loop is terminated
      if(number < 0.0)
      {
         break;
      }
      sum += number; // sum = sum + number;
   }
   printf("Sum = %.2lf",sum);
   return 0;
}
```

continue Statement

The continue statement skips some statements inside the loop. The continue statement is used with decision making statement such as if...else.

```
# include <stdio.h>
int main()
{
   int i;
   double number, sum = 0.0;
   for(i=1; i <= 10; ++i)
   {
      printf("Enter a n%d: ",i);
      scanf("%lf",&number);
      // If user enters negative number, loop is terminated
      if(number < 0.0)
      {
         continue;
      }
      sum += number; // sum = sum + number;
   }
   printf("Sum = %.2lf",sum);
```

```
    return 0;
}
```

AQ 6: Write a Program using switch case statement to create a simple calculator which performs addition, subtraction, multiplication or division depending the input from user

```
# include <stdio.h>

int main() {

    char operator;
    double firstNumber, secondNumber;

    printf("Enter an operator (+, -, *, /): ");
    scanf("%c", &operator);

    printf("Enter two operands: ");
    scanf("%lf %lf", &firstNumber, &secondNumber);

    switch(operator)
    {
      case '+':
        printf("%.1lf + %.1lf = %.1lf",firstNumber, secondNumber, firstNumber + secondNumber);
            break;

      case '-':
           printf("%.1lf - %.1lf = %.1lf",firstNumber, secondNumber, firstNumber - secondNumber);
            break;

      case '*':
        printf("%.1lf * %.1lf = %.1lf",firstNumber, secondNumber, firstNumber * secondNumber);
            break;

       case '/':
           printf("%.1lf / %.1lf = %.1lf",firstNumber, secondNumber, firstNumber / firstNumber);
            break;

       // operator doesn't match any case constant (+, -, *, /)
      default:
            printf("Error! operator is not correct");
    }

    return 0;
}
```