Lab no: 3 – LOOPING CONTROL STRUCTURES – WHILE AND DO LOOPS

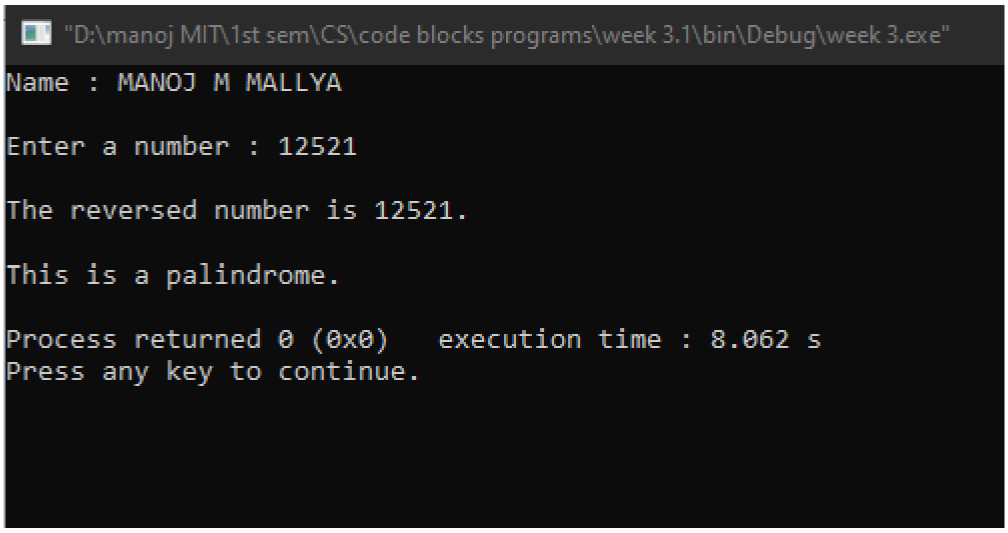
Q1. Reverse a given number and check if it is a palindrome or not. (use while loop). [Ex: 1234, reverse=4*10 3 +3 * 10 2 + 2 * 10 1 + 1 * 10 0 =4321]

Program:

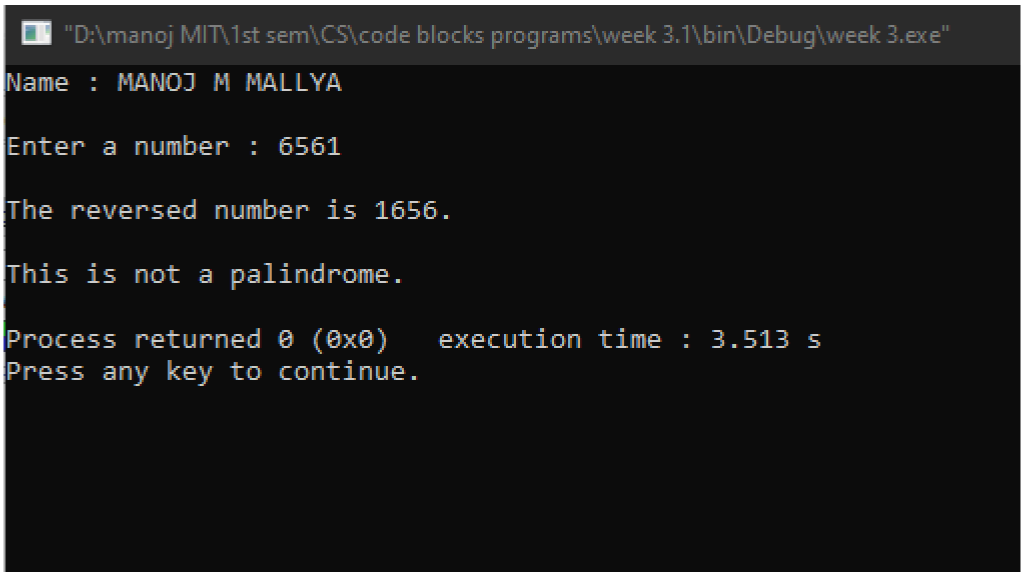
```
//Checking whether a number is palindrome or not
#include <stdio.h>
#include <stdlib.h>

int main()
{
   printf("Name : MANOJ M MALLYA\n\n");
   int num,digit,reverse=0,k;
   printf("Enter a number : ");
   scanf("%d",&num);
   k = num;
   while (num!=0)
   {
      digit = num % 10;
      reverse = reverse*10 + digit ;
      num = num / 10;
   }
   printf("\nThe reversed number is %d.\n",reverse);
   if(k==reverse)
   {
      printf("\nThis is a palindrome.\n");
   }
```

```
        else
        {
            printf("\nThis is not a palindrome.\n");
        }
        return 0;
    }
```

Output:



```
"D:\manoj MIT\1st sem\CS\code blocks programs\week 3.1\bin\Debug\week 3.exe"
Name : MANOJ M MALLYA

Enter a number : 12521

The reversed number is 12521.

This is a palindrome.

Process returned 0 (0x0)   execution time : 8.062 s
Press any key to continue.
```



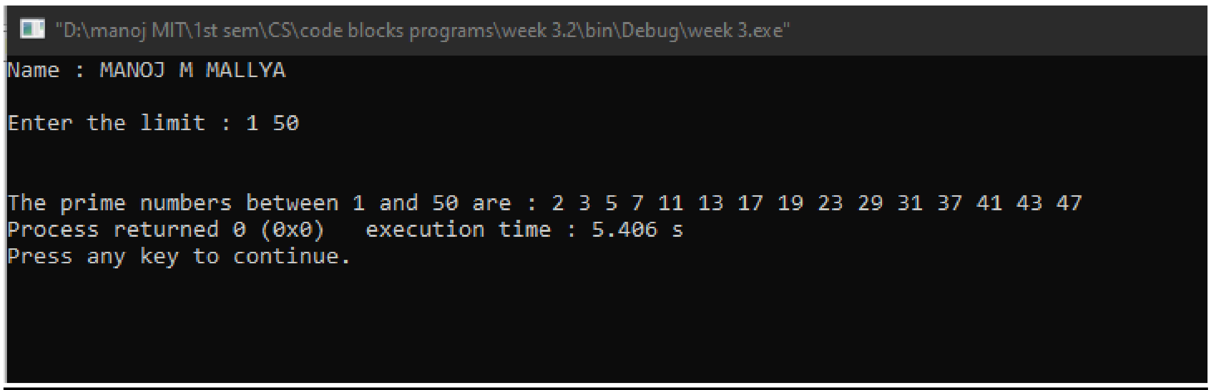
```
"D:\manoj MIT\1st sem\CS\code blocks programs\week 3.1\bin\Debug\week 3.exe"
Name : MANOJ M MALLYA

Enter a number : 6561

The reversed number is 1656.

This is not a palindrome.

Process returned 0 (0x0)   execution time : 3.513 s
Press any key to continue.
```

Q2. Generate prime numbers between 2 given limits.(use while loop)

Program:

```
//Generating prime numbers between given limits
#include <stdio.h>
#include <stdlib.h>

int main()
{
    printf("Name : MANOJ M MALLYA\n\n");
    int start,end,num,flag;
    printf("Enter the limit : ");
    scanf("%d %d",&start,&end);

    printf("\n\nThe prime numbers between %d and %d are : ",start,end);
    while(start<=end)
    {
        flag=1;
        num=2;
        while(num<start)
        {
            if ((start%num)==0)
            {
                flag=0;
                break;
            }
            num++;
        }
        if(flag==1)
        {
            if (start!=1)
            {
                printf("%d ",start);
            }
        }
        start++;
    }
    return 0;
}
```

<u>Output:</u>



"D:\manoj MIT\1st sem\CS\code blocks programs\week 3.2\bin\Debug\week 3.exe"

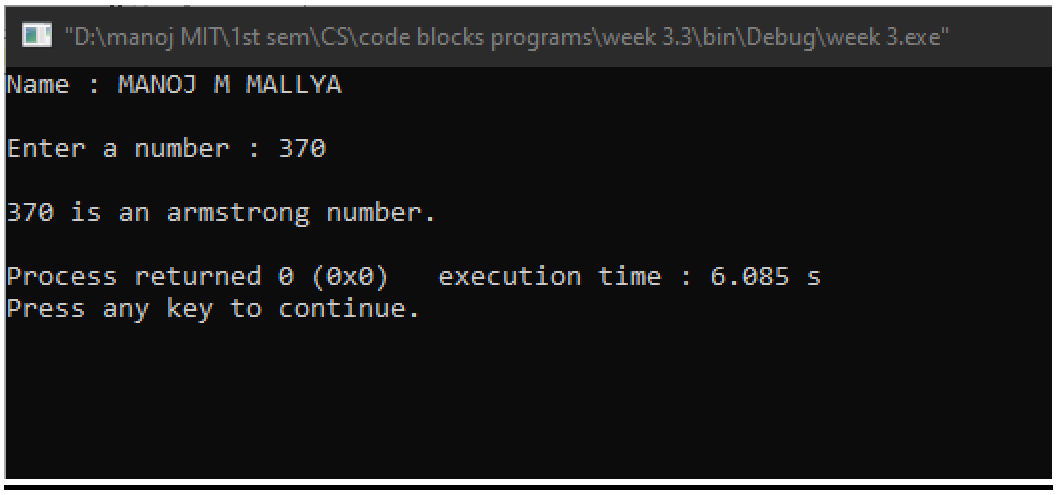
```
Name : MANOJ M MALLYA

Enter the limit : 1 50


The prime numbers between 1 and 50 are : 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47
Process returned 0 (0x0)   execution time : 5.406 s
Press any key to continue.
```

Q3. Check if the sum of the cubes of all digits of an inputted number equals the number itself (Armstrong Number). (use while loop)

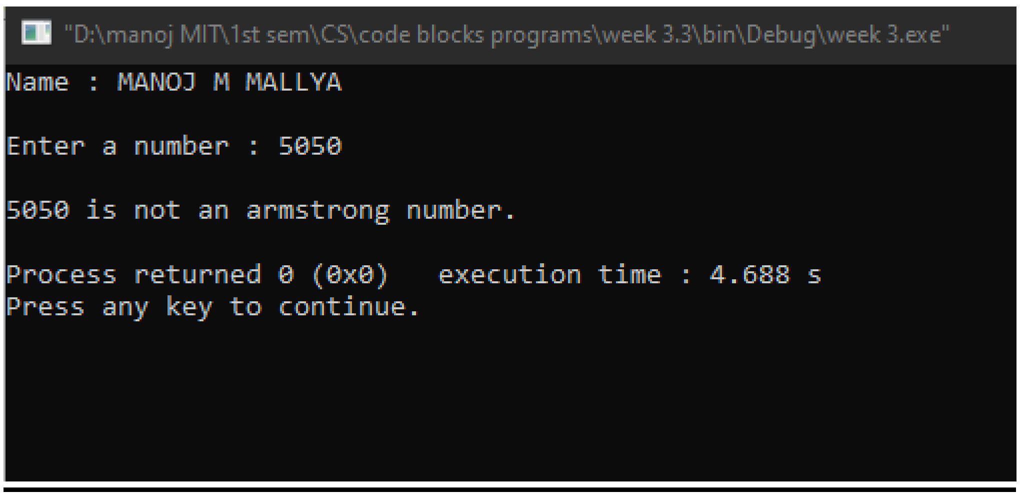
<u>Program:</u>

```
//Checking whether a number is armstrong number or not
#include <stdio.h>
#include <stdlib.h>


int main()
{
    printf("Name : MANOJ M MALLYA\n\n");
    int n,copy,digit,sum=0;
    printf("Enter a number : ");
    scanf("%d",&n);
    copy = n;

    while (n>0)
    {
        digit = n % 10;
        sum = sum + digit * digit * digit ;
        n = n/10;
    }

    if(sum==copy)
    {
        printf("\n%d is an armstrong number.\n",copy);
    }
    else
    {
        printf("\n%d is not an armstrong number.\n",copy);
    }
    return 0;
}
```

Output:



```
"D:\manoj MIT\1st sem\CS\code blocks programs\week 3.3\bin\Debug\week 3.exe"
Name : MANOJ M MALLYA

Enter a number : 370

370 is an armstrong number.

Process returned 0 (0x0)   execution time : 6.085 s
Press any key to continue.
```



```
"D:\manoj MIT\1st sem\CS\code blocks programs\week 3.3\bin\Debug\week 3.exe"
Name : MANOJ M MALLYA

Enter a number : 5050

5050 is not an armstrong number.

Process returned 0 (0x0)   execution time : 4.688 s
Press any key to continue.
```

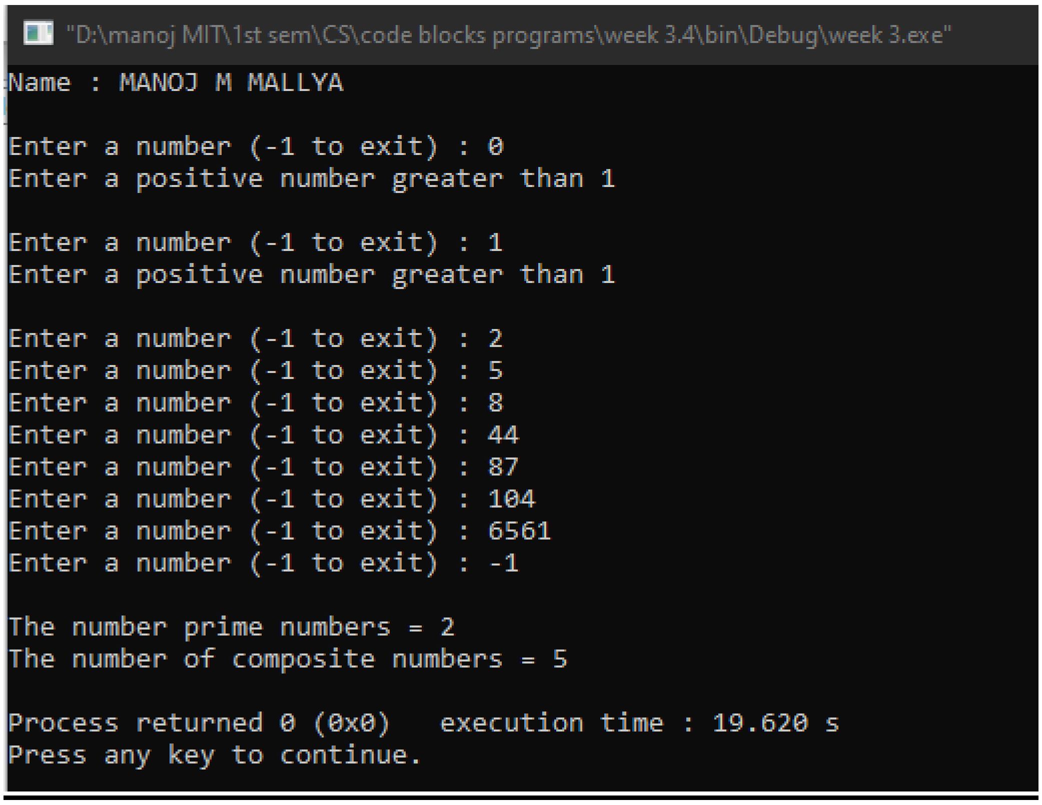
Q4. Write a program using do-while loop to read the numbers until -1 is encountered. Also count the number of prime numbers and composite numbers entered by user. [Hint: 1 is neither prime nor composite]

Program:

```
//Counting the number of prime and composite numbers until -1 is encountered

#include <stdio.h>

#include <stdlib.h>

#include <math.h>


int main()

{

   printf("Name : MANOJ M MALLYA\n\n");
```

```
int n,count_p=0,count_c=0,num,flag;
do
{
    printf("Enter a number (-1 to exit) : ");
    scanf("%d",&n);

    if(n==(-1))
    {
        break;
    }
    if (n>1)
    {
        flag=1;
        num=2;
        do
        {
            if(n%num==0&&n!=2)
            {
                flag=0;
                break;
            }
            num++;
        }
        while(num<=sqrt(n));
        if (flag==1)
        {
            count_p++;
            }
            else
            {
                count_c++;
            }
        }
        else
            printf("Enter a positive number greater than 1\n\n");
    }
    while(1);
    printf("\nThe number prime numbers = %d \nThe number of composite
numbers = %d\n",count_p,count_c);
    return 0;
}
```

Output:



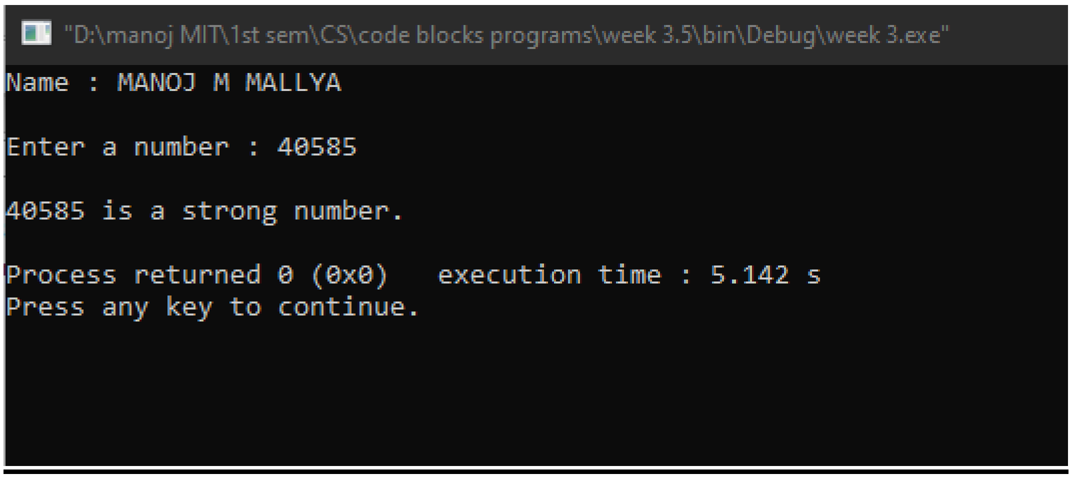
```
"D:\manoj MIT\1st sem\CS\code blocks programs\week 3.4\bin\Debug\week 3.exe"
Name : MANOJ M MALLYA

Enter a number (-1 to exit) : 0
Enter a positive number greater than 1

Enter a number (-1 to exit) : 1
Enter a positive number greater than 1

Enter a number (-1 to exit) : 2
Enter a number (-1 to exit) : 5
Enter a number (-1 to exit) : 8
Enter a number (-1 to exit) : 44
Enter a number (-1 to exit) : 87
Enter a number (-1 to exit) : 104
Enter a number (-1 to exit) : 6561
Enter a number (-1 to exit) : -1

The number prime numbers = 2
The number of composite numbers = 5

Process returned 0 (0x0)   execution time : 19.620 s
Press any key to continue.
```

Q5. Check whether the given number is strong or not. [Hint: Positive number whose sum of the factorial of its digits is equal to the number itself] Ex: 145 = 1! + 4! + 5! = 1 + 24 + 120 = 145 is a strong number.

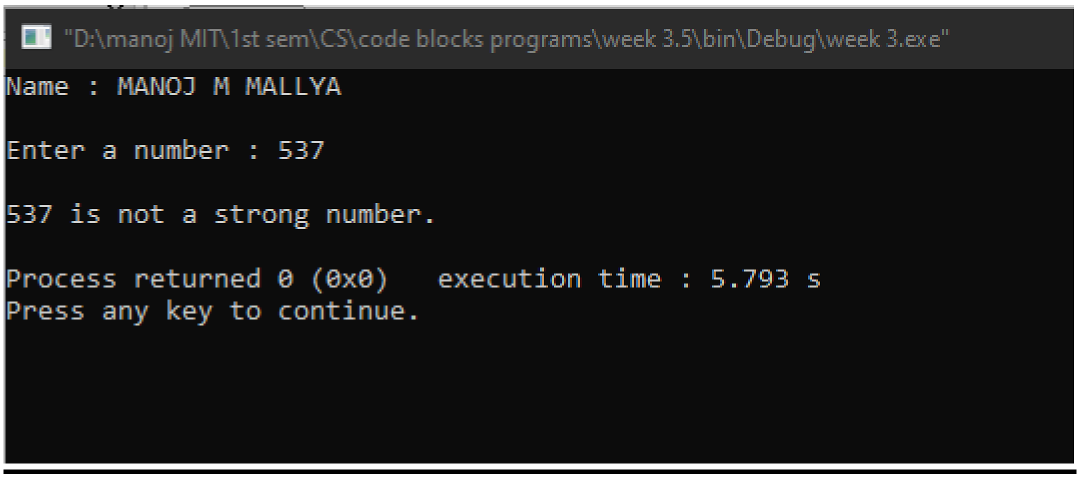
<u>Program:</u>

```
//Checking whether a number is strong number or not
#include <stdio.h>
#include <stdlib.h>

int main()
{
    printf("Name : MANOJ M MALLYA\n\n");
    int n,i,copy,sum=0,fac,digit;
    printf("Enter a number : ");
    scanf("%d",&n);
    copy = n;

    while(copy>0)
    {
        fac=1;
        i=1;
        digit = copy % 10;
        while(i<=digit)
        {
            fac = fac * i;
            i++;
        }
        sum = sum + fac;
        copy=copy/10;
    }

    if(sum==n)
    {
        printf("\n%d is a strong number.\n",n);
    }
    else
    {
        printf("\n%d is not a strong number.\n",n);
    }
    return 0;
}
```

Output:



```
"D:\manoj MIT\1st sem\CS\code blocks programs\week 3.5\bin\Debug\week 3.exe"
Name : MANOJ M MALLYA

Enter a number : 40585

40585 is a strong number.

Process returned 0 (0x0)   execution time : 5.142 s
Press any key to continue.
```



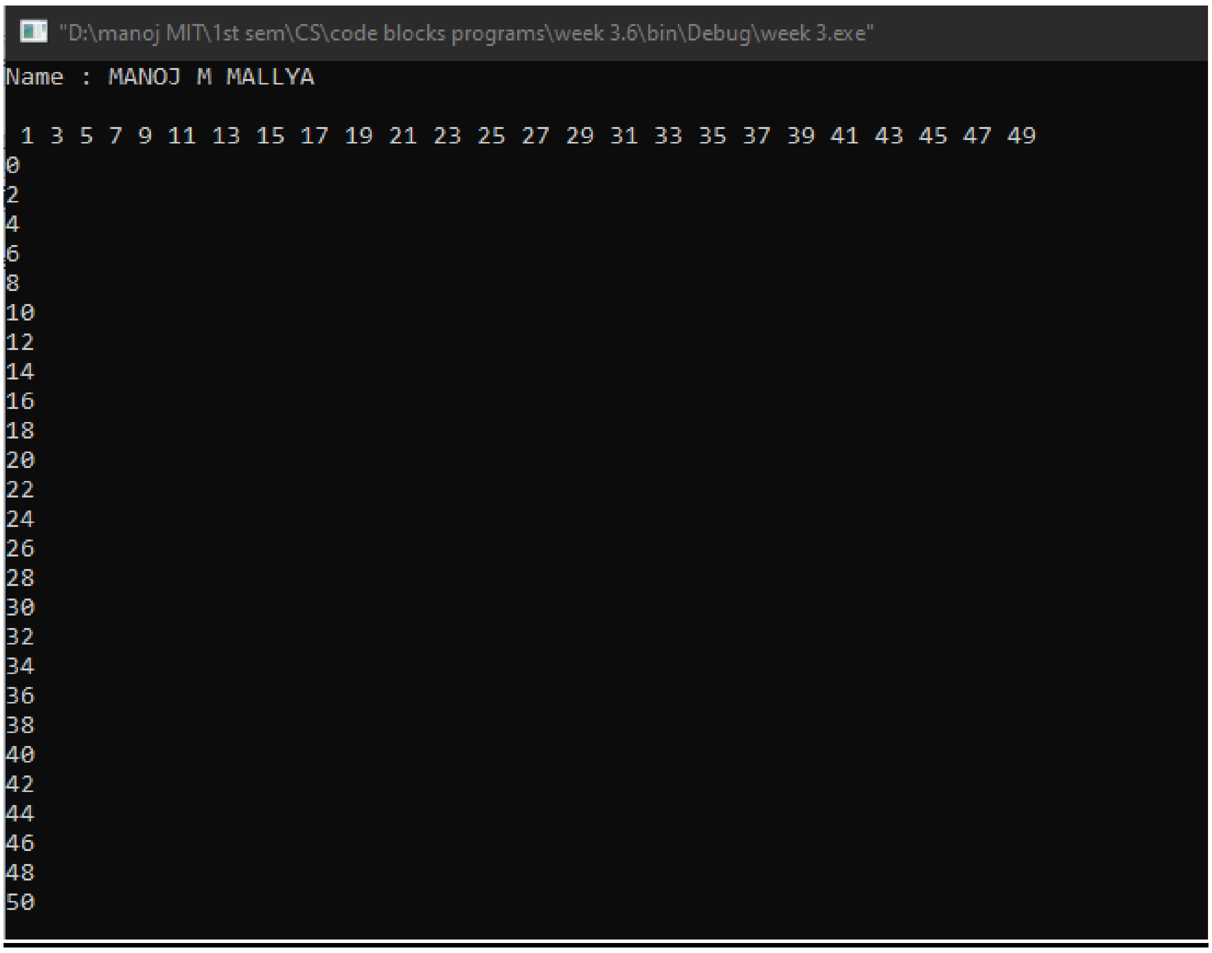
```
"D:\manoj MIT\1st sem\CS\code blocks programs\week 3.5\bin\Debug\week 3.exe"
Name : MANOJ M MALLYA

Enter a number : 537

537 is not a strong number.

Process returned 0 (0x0)   execution time : 5.793 s
Press any key to continue.
```

Q6. Write a program to demonstrate use of break and continue statements in while and do-while loops.

<u>Program:</u>

```
//Printing odd numbers horizontally and even numbers vertically

#include <stdio.h>

#include <stdlib.h>


int main()

{

    printf("Name : MANOJ M MALLYA\n\n");

    int i=0,j=0;

    //printing odd numbers horizontally using while loop

    while (i<=100)

    {

        if(i%2==0)

        {

            i++;

            continue;

        }


        if (i>50)

        {

            break;

        }

        printf(" %d",i);


        i++;
    }

    //printing even numbers vertically using do while loop

    printf("\n");

    do
    {
        if(j%2!=0)
        {
            j++;
            continue;
        }

        if (j>50)
        {
            break;
        }
        printf("%d\n",j);

        j++;
    }
    while(j<=100);

    return 0;
}
```

Output:



"D:\manoj MIT\1st sem\CS\code blocks programs\week 3.6\bin\Debug\week 3.exe"

```
Name : MANOJ M MALLYA

 1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
40
42
44
46
48
50
```

Lab no: 4 – LOOPING CONTROL STRUCTURES – FOR LOOPS

Q1. Generate the multiplication table for ‘n’ numbers up to ‘k’ terms (using nested for loops).

[Hint: 1 2 3 4 5 …. k

2 4 6 8 10 ….2*k

..…

……

……

……

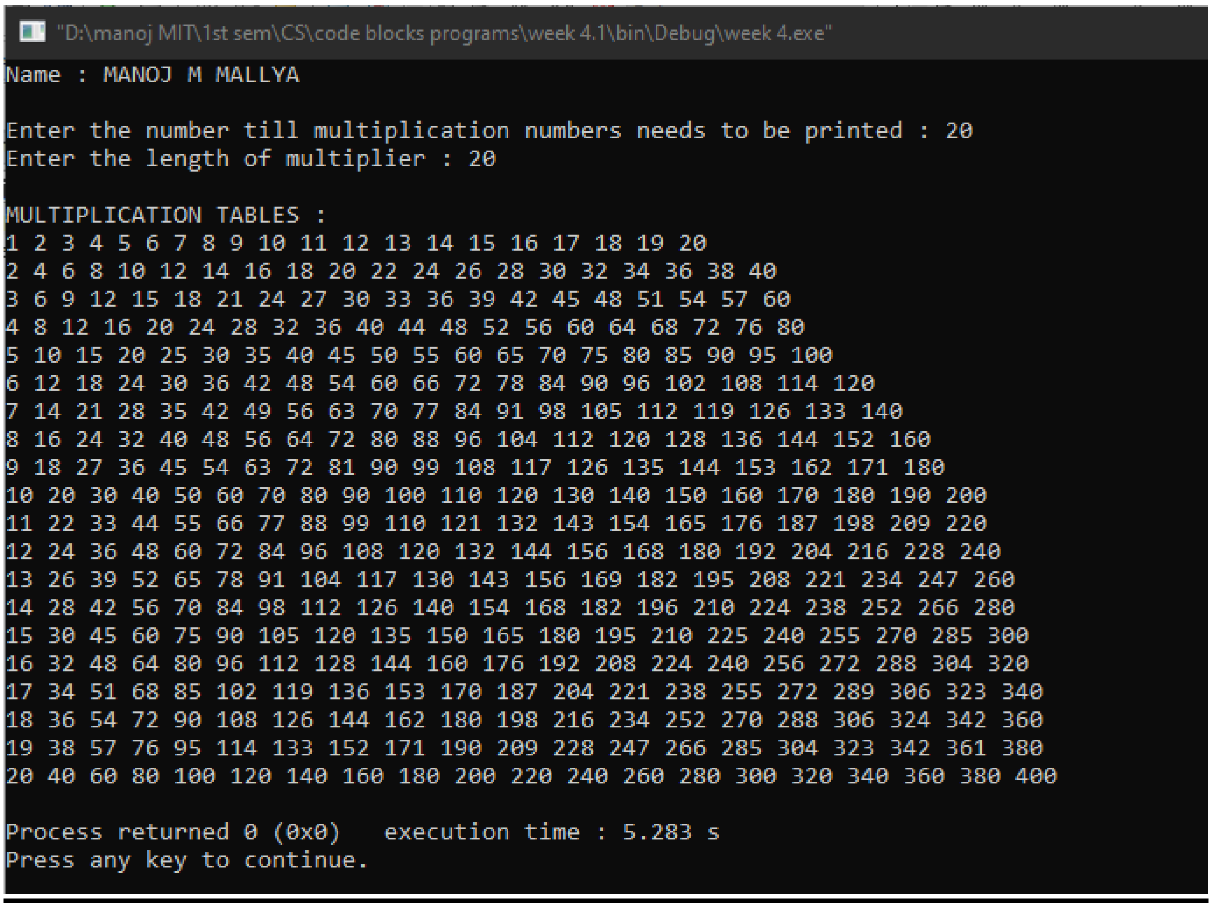
n……………… n*k]

Program:

```
//Generating multiplication table
#include <stdio.h>
#include <stdlib.h>

int main()
{
   printf("Name : MANOJ M MALLYA\n\n");
   int n,k;
   printf("Enter the number till multiplication numbers needs to be printed : ");
   scanf("%d",&n);
   printf("Enter the length of multiplier : ");
   scanf("%d",&k);

   printf("\nMULTIPLICATION TABLES : \n");
```

```
    for(int i=1;i<=n;i++)
    {
        for(int j=1;j<=k;j++)
        {
            printf("%d ",i*j);
        }
        printf("\n");
    }
    return 0;
}
```

Output:



"D:\manoj MIT\1st sem\CS\code blocks programs\week 4.1\bin\Debug\week 4.exe"

```
Name : MANOJ M MALLYA

Enter the number till multiplication numbers needs to be printed : 20
Enter the length of multiplier : 20

MULTIPLICATION TABLES :
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40
3 6 9 12 15 18 21 24 27 30 33 36 39 42 45 48 51 54 57 60
4 8 12 16 20 24 28 32 36 40 44 48 52 56 60 64 68 72 76 80
5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100
6 12 18 24 30 36 42 48 54 60 66 72 78 84 90 96 102 108 114 120
7 14 21 28 35 42 49 56 63 70 77 84 91 98 105 112 119 126 133 140
8 16 24 32 40 48 56 64 72 80 88 96 104 112 120 128 136 144 152 160
9 18 27 36 45 54 63 72 81 90 99 108 117 126 135 144 153 162 171 180
10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180 190 200
11 22 33 44 55 66 77 88 99 110 121 132 143 154 165 176 187 198 209 220
12 24 36 48 60 72 84 96 108 120 132 144 156 168 180 192 204 216 228 240
13 26 39 52 65 78 91 104 117 130 143 156 169 182 195 208 221 234 247 260
14 28 42 56 70 84 98 112 126 140 154 168 182 196 210 224 238 252 266 280
15 30 45 60 75 90 105 120 135 150 165 180 195 210 225 240 255 270 285 300
16 32 48 64 80 96 112 128 144 160 176 192 208 224 240 256 272 288 304 320
17 34 51 68 85 102 119 136 153 170 187 204 221 238 255 272 289 306 323 340
18 36 54 72 90 108 126 144 162 180 198 216 234 252 270 288 306 324 342 360
19 38 57 76 95 114 133 152 171 190 209 228 247 266 285 304 323 342 361 380
20 40 60 80 100 120 140 160 180 200 220 240 260 280 300 320 340 360 380 400

Process returned 0 (0x0)   execution time : 5.283 s
Press any key to continue.
```

Q2. Generate Floyd’s triangle using natural numbers for a given limit N. (using for loops)

[Hint: Floyd’s triangle is a right angled-triangle using the natural numbers] Ex: Input: N = 4

Output: 1

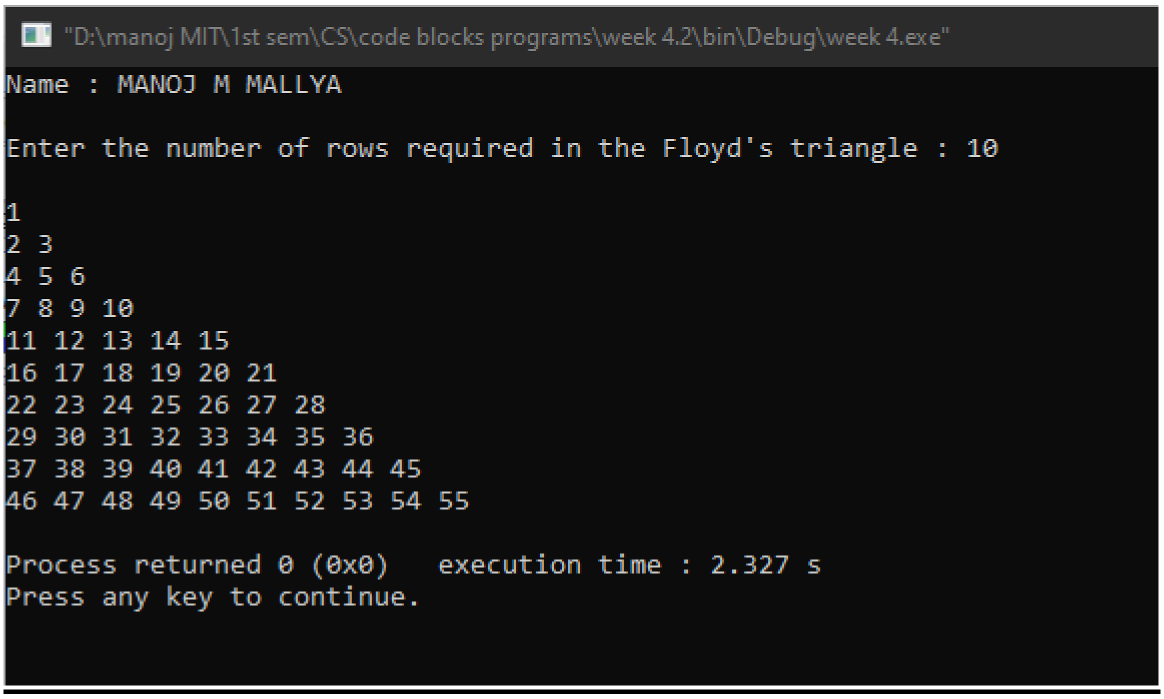
```
        2 3
        4 5 6
        7 8 9 10
```

<u>Program:</u>

```
//Generating Floyd’s triangle using natural numbers for a given limit N
#include <stdio.h>
#include <stdlib.h>

int main()
{
   printf("Name : MANOJ M MALLYA\n\n");
   int N,k=1;
   printf("Enter the number of rows required in the Floyd's triangle : ");
   scanf("%d",&N);

   printf("\n");
   for(int i=0;i<N;i++)
   {
      for(int j=N-i;j<=N;j++)
      {
         printf("%d ",k);
         k++;
      }
```

```
        printf("\n");
    }
    return 0;
}
```

Output:



"D:\manoj MIT\1st sem\CS\code blocks programs\week 4.2\bin\Debug\week 4.exe"

```
Name : MANOJ M MALLYA

Enter the number of rows required in the Floyd's triangle : 10

1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31 32 33 34 35 36
37 38 39 40 41 42 43 44 45
46 47 48 49 50 51 52 53 54 55

Process returned 0 (0x0)   execution time : 2.327 s
Press any key to continue.
```

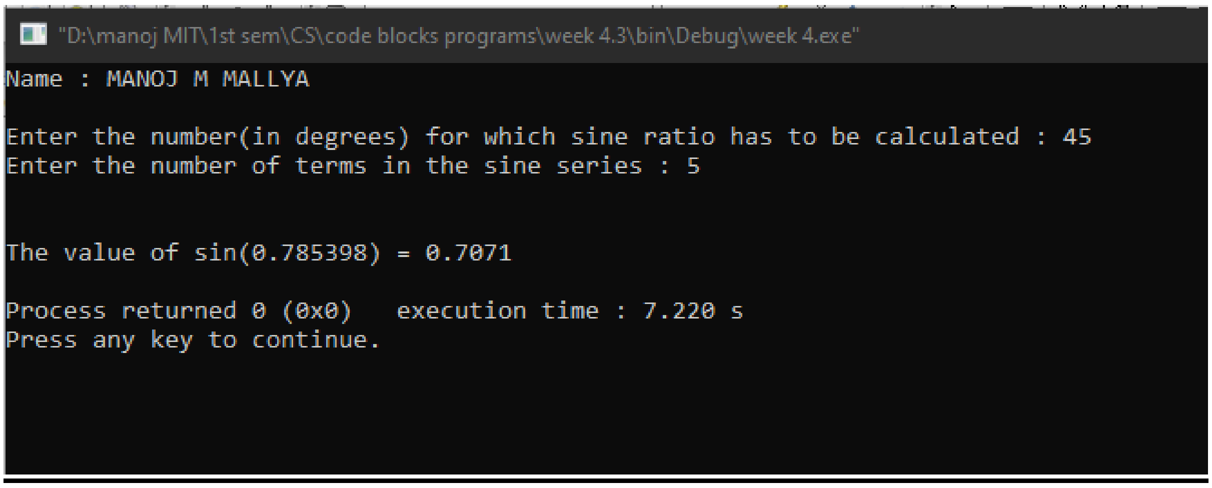
Q3. Evaluate the sine series, sin(x)= x- x^3 /3! + x^5 /5! – x^7 /7! + ……… to n terms.

Program:

```
//Evaluating sine of an angle using infinite series expansion(limited to an input number)

#include <stdio.h>

#include <stdlib.h>

#define pi 3.14159265


int main()
```

```
{
   printf("Name : MANOJ M MALLYA\n\n");
   int i,n;
   float x,sum,k;

   printf("Enter the number(in degrees) for which sine ratio has to be calculated
: ");
   scanf("%f",&x);
   printf("Enter the number of terms in the sine series : ");
   scanf("%d",&n);

   x = (x * pi)/180;
   k = x;
   sum = x;

   //logical for loop for the sine series generation
   for(i=1;i<=n;i++)
   {
      k = (k*(-1)*x*x)/(2*i*(2*i+1));
      sum = sum + k;
   }

   printf("\n\nThe value of sin(%f) = %.4f\n",x,sum);
   return 0;
}
```

Output:



```
"D:\manoj MIT\1st sem\CS\code blocks programs\week 4.3\bin\Debug\week 4.exe"
Name : MANOJ M MALLYA

Enter the number(in degrees) for which sine ratio has to be calculated : 45
Enter the number of terms in the sine series : 5


The value of sin(0.785398) = 0.7071

Process returned 0 (0x0)   execution time : 7.220 s
Press any key to continue.
```

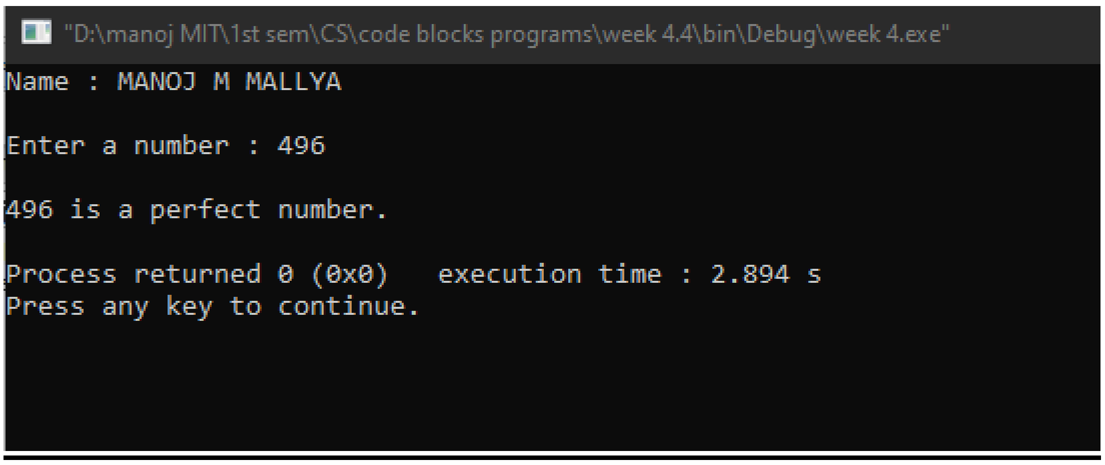
Q4. Check whether a given number is perfect or not.

[Hint: Sum of all positive divisors of a given number excluding the given number is equal to the number] Ex: 28 = 1+ 2 + 4 + 7 + 14 = 28 is a perfect number

Program:

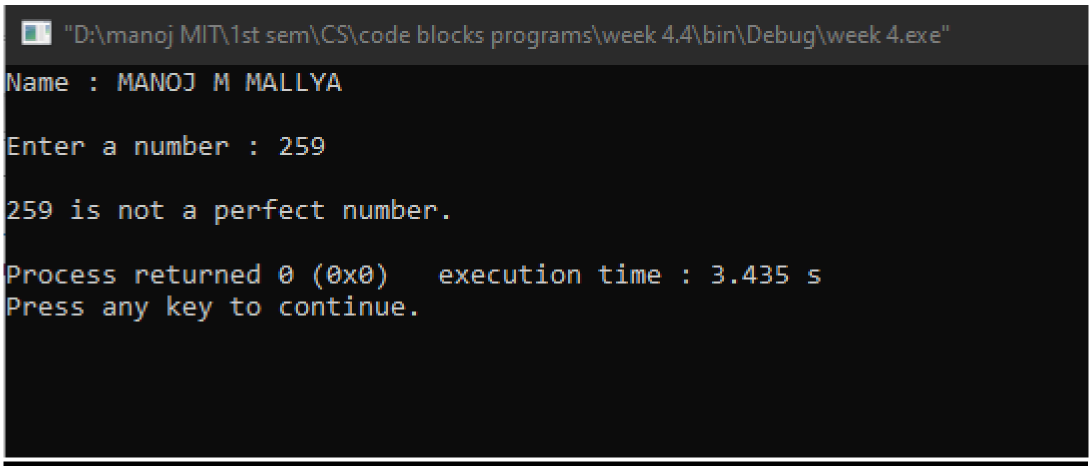
```
//Checking whether a given number is perfect or not

#include <stdio.h>

#include <stdlib.h>


int main()

{

    printf("Name : MANOJ M MALLYA\n\n");

    int i,n,sum=0;

    printf("Enter a number : ");

    scanf("%d",&n);


    for(i=1;i<n;i++)
    {
        if(n%i==0)
        {
            sum+=i;
        }
    }

    if(sum==n)
    {
        printf("\n%d is a perfect number.\n",n);
    }
    else
    {
        printf("\n%d is not a perfect number.\n",n);
    }
    return 0;
}
```

<u>Output:</u>



"D:\manoj MIT\1st sem\CS\code blocks programs\week 4.4\bin\Debug\week 4.exe"

```
Name : MANOJ M MALLYA

Enter a number : 496

496 is a perfect number.

Process returned 0 (0x0)   execution time : 2.894 s
Press any key to continue.
```


"D:\manoj MIT\1st sem\CS\code blocks programs\week 4.4\bin\Debug\week 4.exe"

```
Name : MANOJ M MALLYA

Enter a number : 259

259 is not a perfect number.

Process returned 0 (0x0)   execution time : 3.435 s
Press any key to continue.
```

Q5. Find out the generic root of any number.

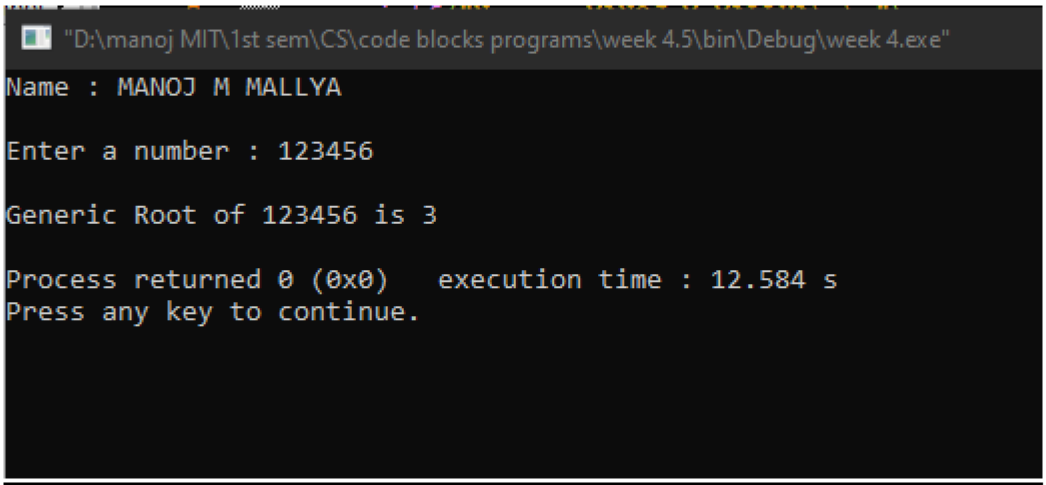
[Hint: Generic root is the sum of digits of a number until a single digit is obtained.] Ex: Generic root of 456 is 4 + 5 + 6 = 15 = 1+5 = 6

Program:

```
//Finding out the generic root of a given number

#include <stdio.h>

#include <stdlib.h>


int main()

{

   printf("Name : MANOJ M MALLYA\n\n");

   int n,temp,rem,sum;


   printf("Enter a number : ");

   scanf("%d",&n);


   temp = n;


   for(;temp>0;)
```

```
    {
       for(sum=0; temp>0;temp=temp/10)
       {
          rem = temp%10;
          sum = sum + rem;
       }

          if(sum>9)
          {
             temp = sum;
          }
          else
          {
             break;
          }
    }
    printf("\nGeneric Root of %d is %d\n", n, sum);
    return 0;
}
```

Output:



"D:\manoj MIT\1st sem\CS\code blocks programs\week 4.5\bin\Debug\week 4.exe"

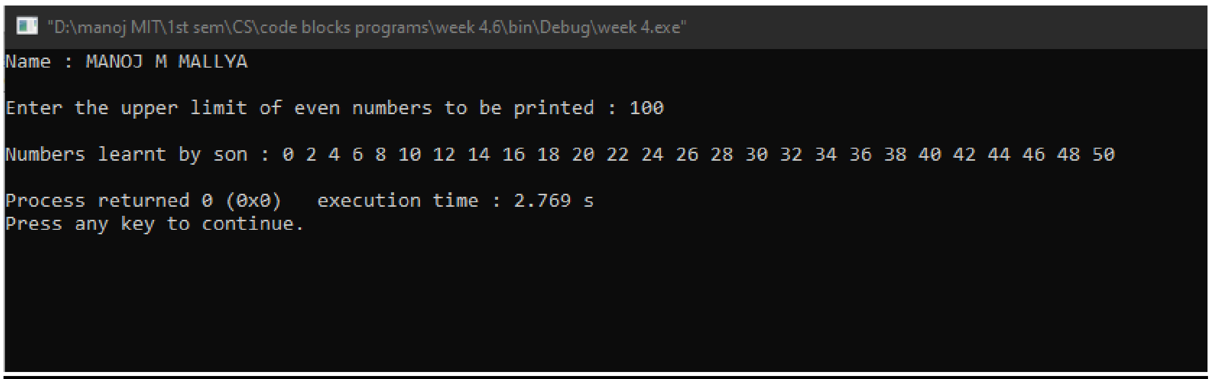
```
Name : MANOJ M MALLYA

Enter a number : 123456

Generic Root of 123456 is 3

Process returned 0 (0x0)   execution time : 12.584 s
Press any key to continue.
```

Q6. Write a program to demonstrate use of break and continue statements in for loop.

<u>**Program:**</u>

```
/*father tells his kindergarten son to learn even numbers upto a certain limit;
but this adamant boy learns only till half of the limit*/

#include <stdio.h>
#include <stdlib.h>

int main()
{
   printf("Name : MANOJ M MALLYA\n\n");
   int n;
   printf("Enter the upper limit of even numbers to be printed : ");
   scanf("%d",&n);

   printf("\n");
   printf("Numbers learnt by son : ");
   for(int i=0;i<=n;i++)
   {
      if(i%2==0)
      {
         printf("%d ",i);
      }
      else
      {
            continue;
        }

        if(i>=n/2)
        {
            break;
        }
    }
    printf("\n");
    return 0;
}
```

Output:



```
"D:\manoj MIT\1st sem\CS\code blocks programs\week 4.6\bin\Debug\week 4.exe"
Name : MANOJ M MALLYA

Enter the upper limit of even numbers to be printed : 100

Numbers learnt by son : 0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50

Process returned 0 (0x0)   execution time : 2.769 s
Press any key to continue.
```