**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

{

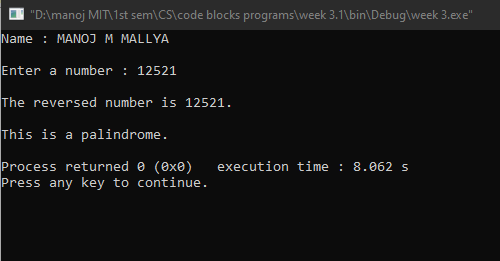
printf("\nThis is not a palindrome.\n");

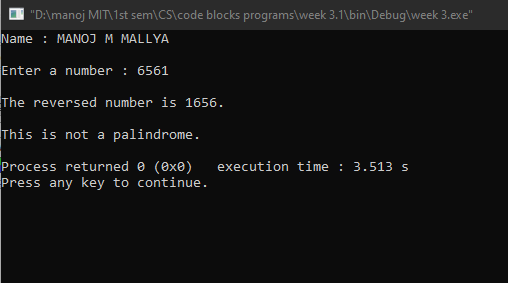
}

return 0;

}

**Output:**





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);

}

}

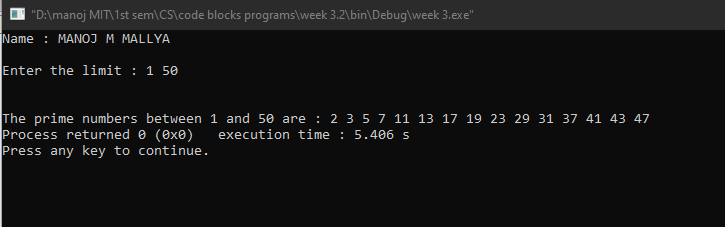
start++;

}

return 0;

}

**Output:**

****

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

**Program:**

*//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

{

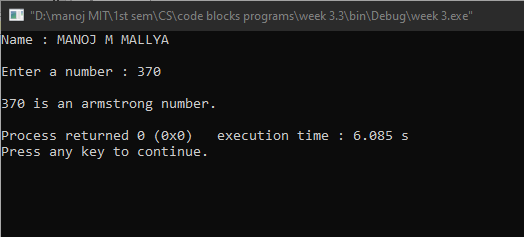
printf("\n%d is not an armstrong number.\n",copy);

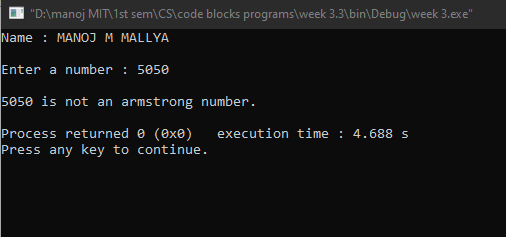
}

return 0;

}

**Output:**

****

****

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");

}

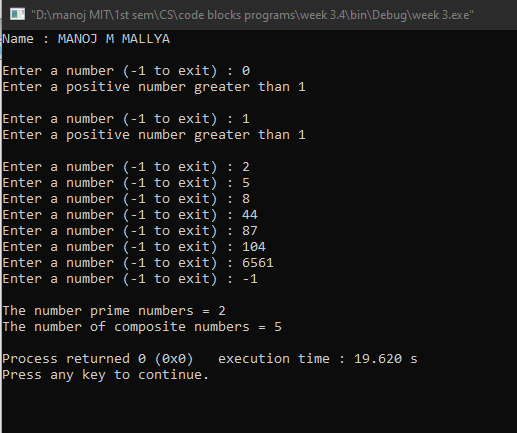
while(1);

printf("\nThe number prime numbers = %d \nThe number of composite numbers = %d\n",count\_p,count\_c);

return 0;

}

**Output:**

****

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.

**Program:**

*//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

{

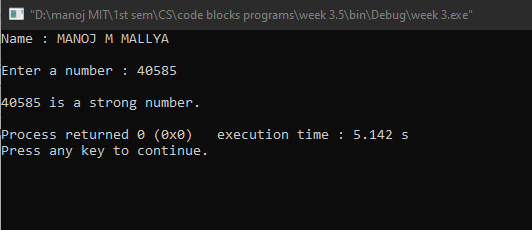
printf("\n%d is not a strong number.\n",n);

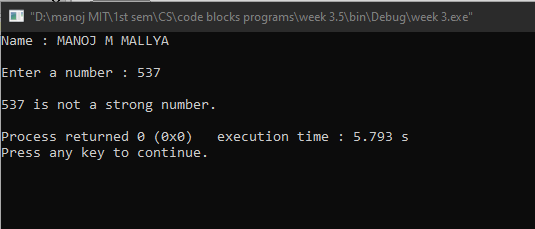
}

return 0;

}

**Output:**

****

****

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

**Program:**

*//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++;

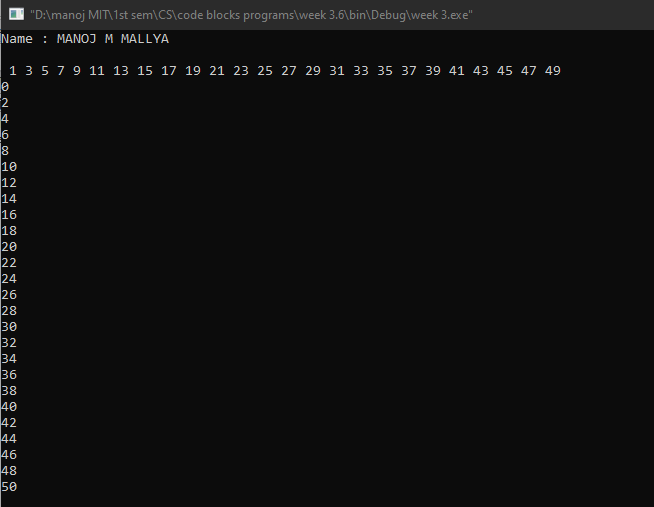
}

while(j<=100);

return 0;

}

**Output:**

****

**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);

}

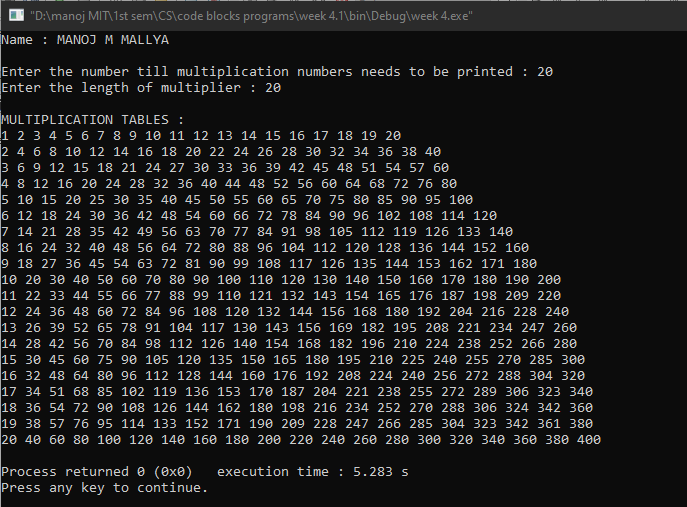
printf("\n");

}

return 0;

}

**Output:**

****

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

**Program:**

*//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++;

}

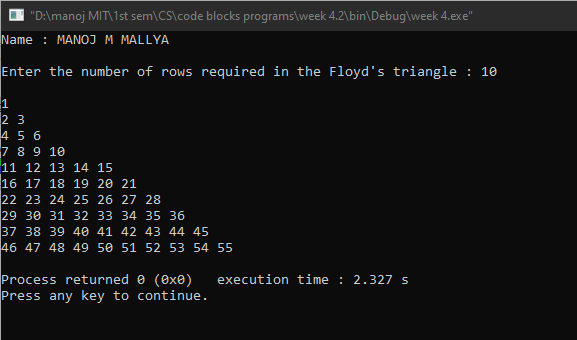
printf("\n");

}

return 0;

}

**Output:**

****

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;

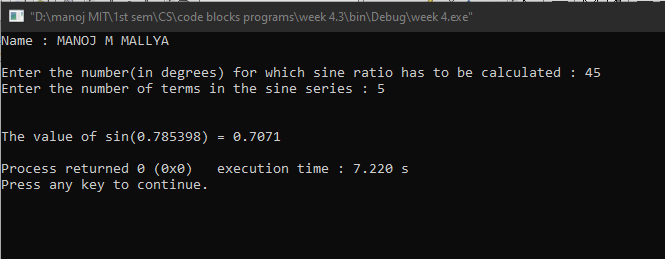
}

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

return 0;

}

**Output:**

****

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

{

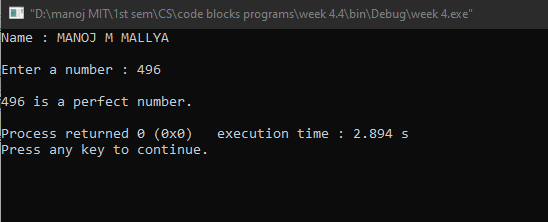
printf("\n%d is not a perfect number.\n",n);

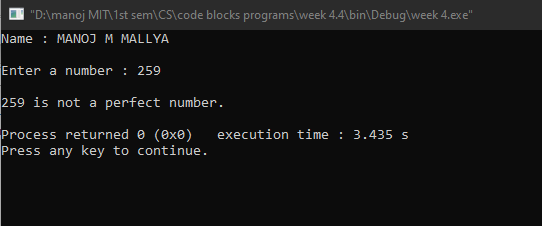
}

return 0;

}

**Output:**

****

****

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;

}

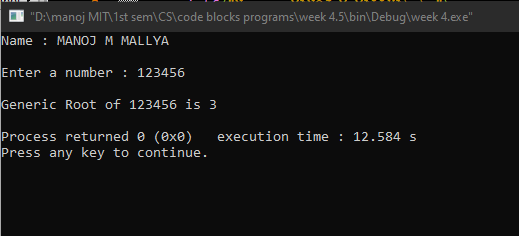
}

printf("\nGeneric Root of %d is %d\n", n, sum);

return 0;

}

**Output:**

****

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

**Program:**

*/\*****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;

}

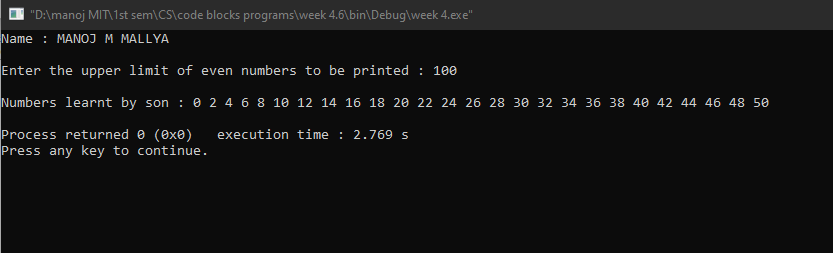
}

printf("\n");

return 0;

}

**Output:**

****