INDEX					
MODULE NO	Date	IDX	PROGRAM	PAGE NO	Sign & Marks
6		**Loops, while and for loops ---- Iterative problems**			
		6.1	Find the given number is a prime or not		
		6.2	Count No of Digits		
		6.3	Checking a number palindrome		
		6.4 **6.5**	Construct a pyramid of numbers 1 * 2 2 * * 3 3 3 * * * 4 4 4 4 * * * * 5 5 5 5 5 * * * * *		
		6.6	Reverse the given no		
		6.7	Fibonacci series		
		6.8	Armstrong or not		
		6.9	read prices of items until decide to stop. print the total cost up to two decimal points.		
		6.10	Find two numbers are co-primes or not.		

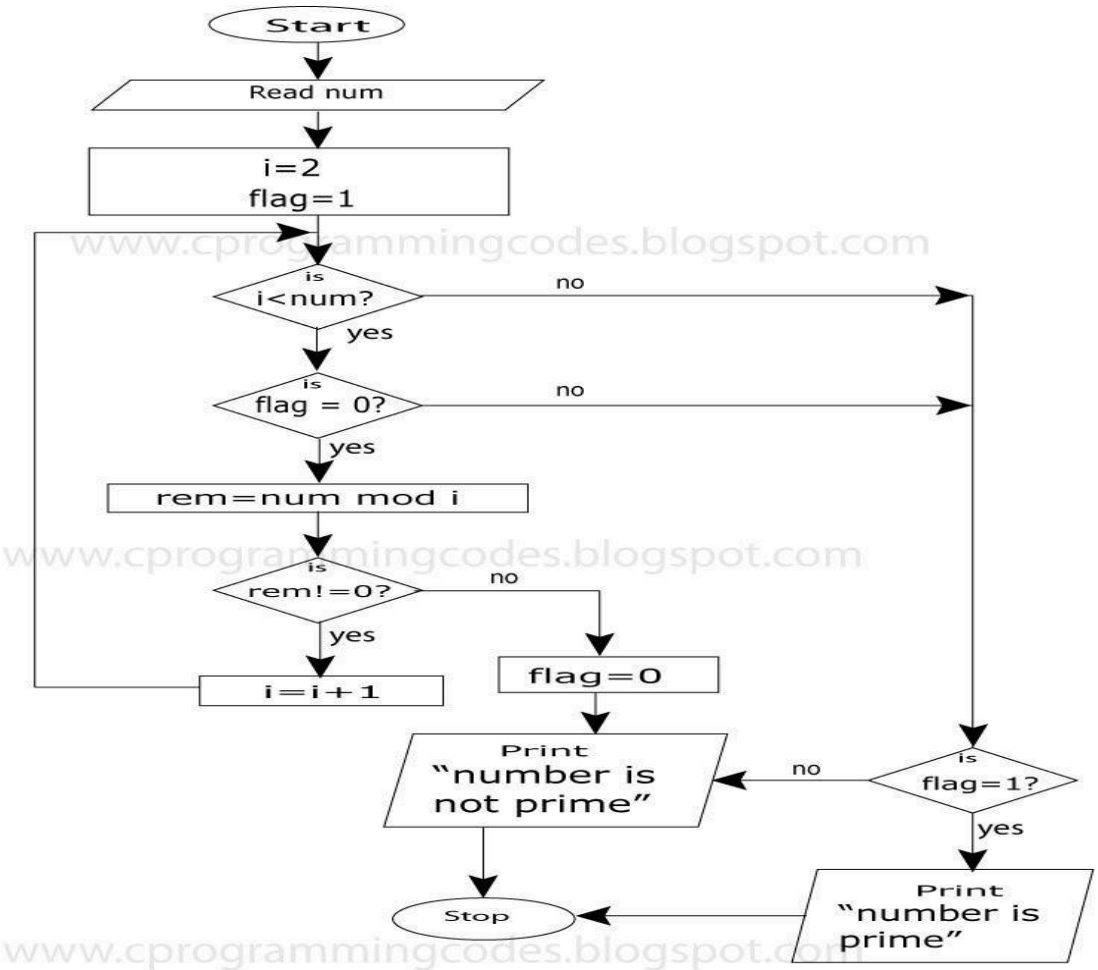
MODULE 6 - Program 6.1

AIM : Write a C Program to find whether the given a number is prime or not.

ALGORITHM :

Step 1: start
Step 2: read n
Step 3: i=2
Step 4: repeat step 5 to 7 until i<=n/2
Step 5: put r=n%i
Step 6: if (r==0) then display not prime goto 8
Step7 : i=i+1
Step 7: if (i>n/2) display it is a prime
Step 8: stop

FLOWCHART:



PROGRAM :

```
#include<stdio.h>
int main()
 {
  int n,r,i=1;
  printf("\n enter the number \t");
  scanf("%d",&n);
  for(i=2; i<=n/2; i++)
   {
    r= n%i;
    if(r==0)
     break;
   }
   if(i>n/2)
     printf("\n it is a prime \n");
   else
     printf("\n it is not a prime \n");

  return 0;
 }
```

OUTPUT :

```
enter the number       5
 it is a prime

 enter the number       21
 it is not a prime
```

MODULE 6 – PROGRAM 6.2

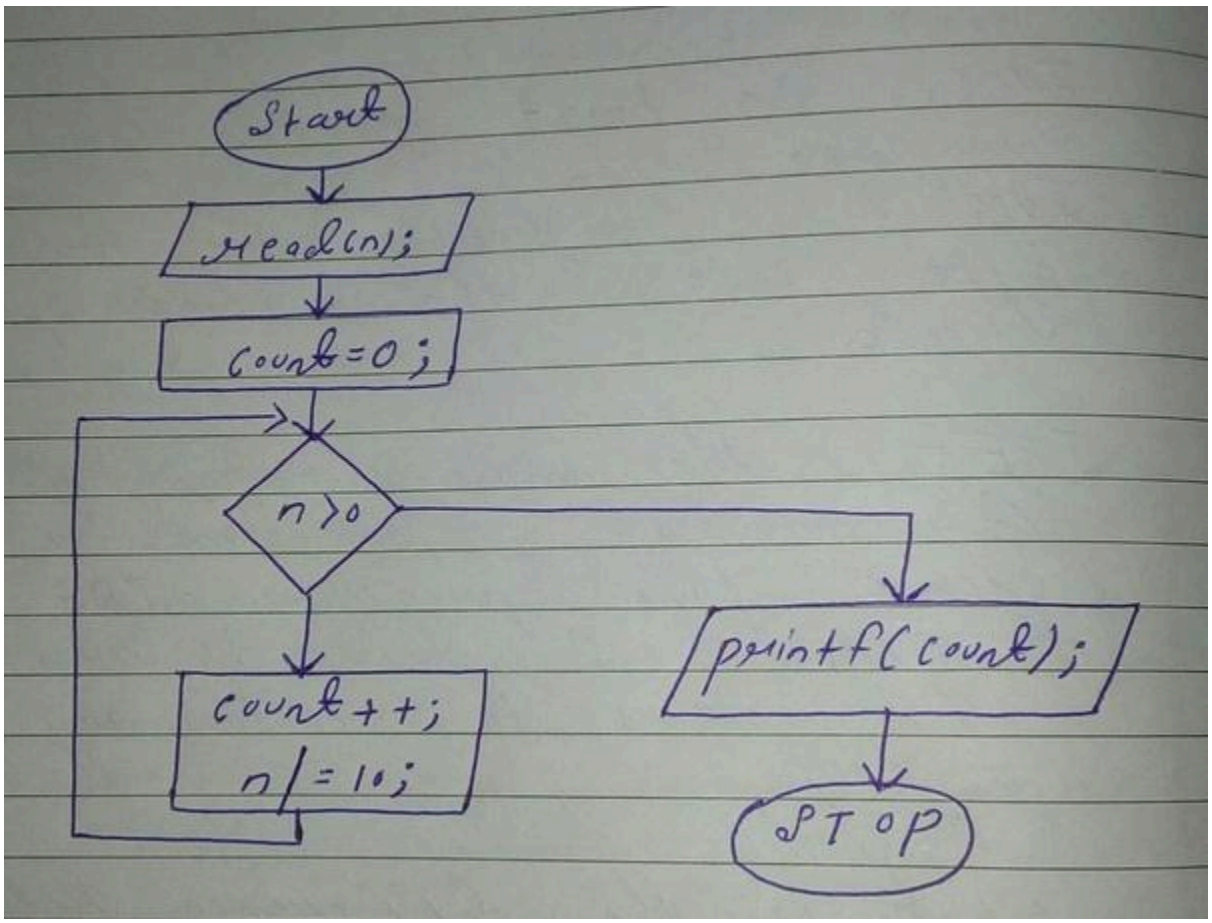
AIM : write a C program to count no of digits of a given number.

ALGORITHM :

STEP 1 : START

STEP 2 : Read a num .
STEP 3 : count = 0.
STEP 4 : If num > 0 then increment count by 1 i.e. count++.
STEP 5 : Divide num by 10 i.e., num= num / 10.
STEP 6 : Repeat step 4 to 5 till num > 0 or num != 0.
STEP 7 : Display count
STEP 8 : STOP

FLOWCHART :



PROGRAM :

```
#include <stdio.h>
int main()
{
   int num, count=0;
   printf("Enter any number: ");
   scanf("%d", &num);
   do
   {
      count++;
      num /= 10;
   } while(num != 0);
   printf("Total digits: %d", count);
   return 0;
}
```

OUTPUT :

```
Enter any number: 1234
Total digits: 4
```

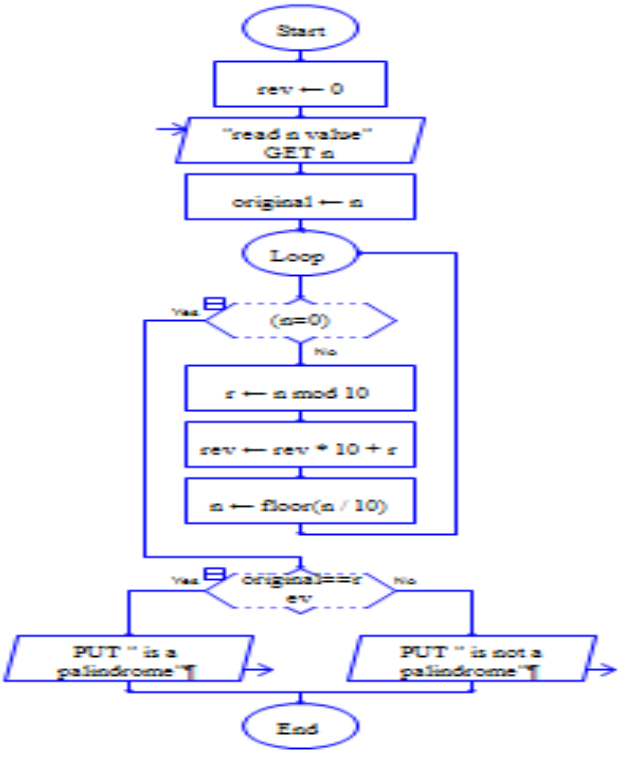
MODULE 6 – PROGRAM 6.3

AIM : write a C program find a number is palindrome (or) not.

ALGORITHM :

STEP 1 : start
STEP 2 : read a number
STEP 3 : [initialize] rev=0,originalvalue=n
STEP 4 : repeat step 4 through 7 until (n!=0)
STEP 5 : r=n%10
STEP 6 : rev=rev*10+r
STEP 7 : n=n/10
STEP 8 : if(originalvalue==rev)
Display "number is palindrome"
otherwise
Display"number not a palindrome"
STEP 9 : stop

FLOWCHART:



PROGRAM :

```
#include<stdio.h>
int main()
{
  int n,r,rev=0,m;
  printf("enter the number \t");
  scanf("%d",&n);
  m = n;
  while(n>0)
  {
   r = n%10;
   rev = rev*10 + r;
   n = n/10;
  }
   printf(" \nreverse of  a number = %d\n",rev);
   if( m == rev )
     printf(" palendrome");
   else
     printf("not a palendrome");
return 0;
}
```

OUTPUT :

```
enter the number      121
reverse of  a number = 121
 palendrome

 enter the number       123
reverse of  a number = 321
not a palendrome
```

MODULE 6 – PROGRAM 6.4

AIM : Write a C Program to print the following format.

1
2 2
3 3 3
4 4 4 4 4
5 5 5 5 5

ALGORITHM :

Step 1 : start
Step 2 : read the number of lines n
Step3 : Taking i from 1 to n do step 4
Step4 : varying j from 1 to i display j
step6 : Stop

PROGRAM :

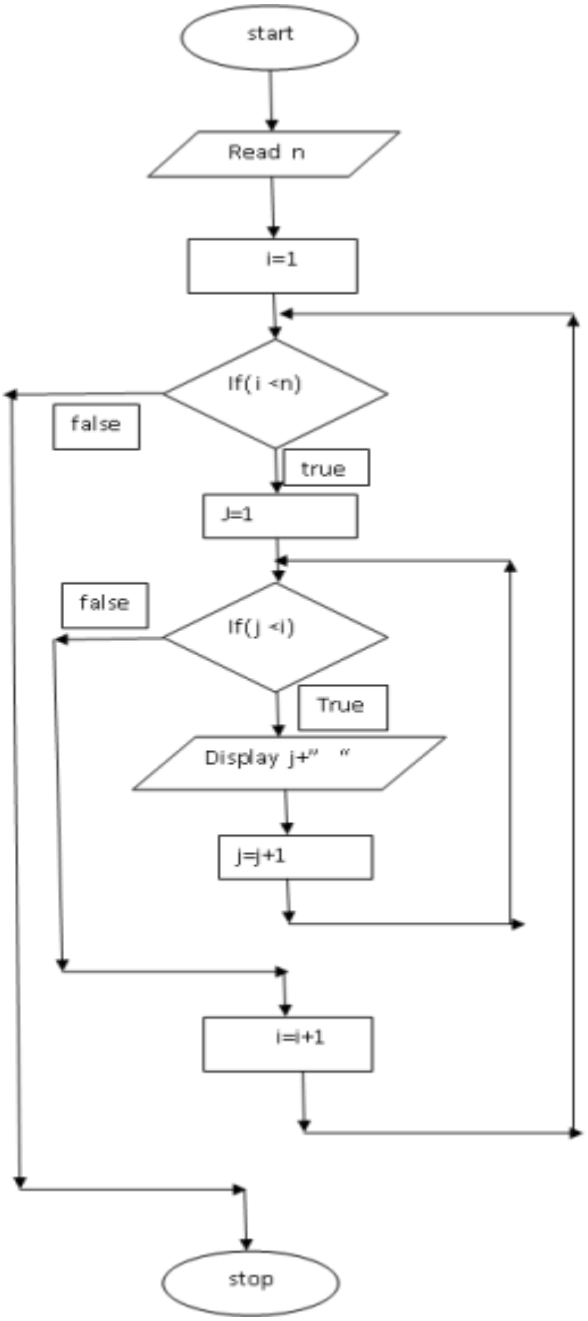
```
#include<stdio.h>
int main()
 {
  int n,i=1,j;
  printf("\n enter the number of lines \t");
  scanf("%d",&n);
  while(i<=n)
  {
   j=1;
   while(j<=i)
   {
    printf("%d\t",i);
    j++;
   }
   printf("\n\n");
   i++;
  }
return 0;
}
```

OUTPUT :

enter the number of lines 5

```
1
2  2
3  3  3
4  4  4  4
5  5  5  5  5
```

FLOWCHART:



MODULE 6 – PROGRAM 6.5

AIM : Write a C Program to print the following format.

```
enter no. of rows: 5
        *
      * * *
    * * * * *
  * * * * * * *
* * * * * * * * *
```

ALGORITHM :

Step 1 : Decide the number of rows and columns

There is a typical structure to print any pattern, i.e., the number of rows and columns. We need to use two loops to print any pattern, i.e., use nested loops.

The outer loop tells us the number of rows, and the inner loop tells us the column needed to print the pattern.

Accept the number of rows from a user **Iterate rows**

Step 2 : Iterate columns

Next, write the inner loop or nested loop to handle the number of columns. The internal loop iteration depends on the values of the outer loop.

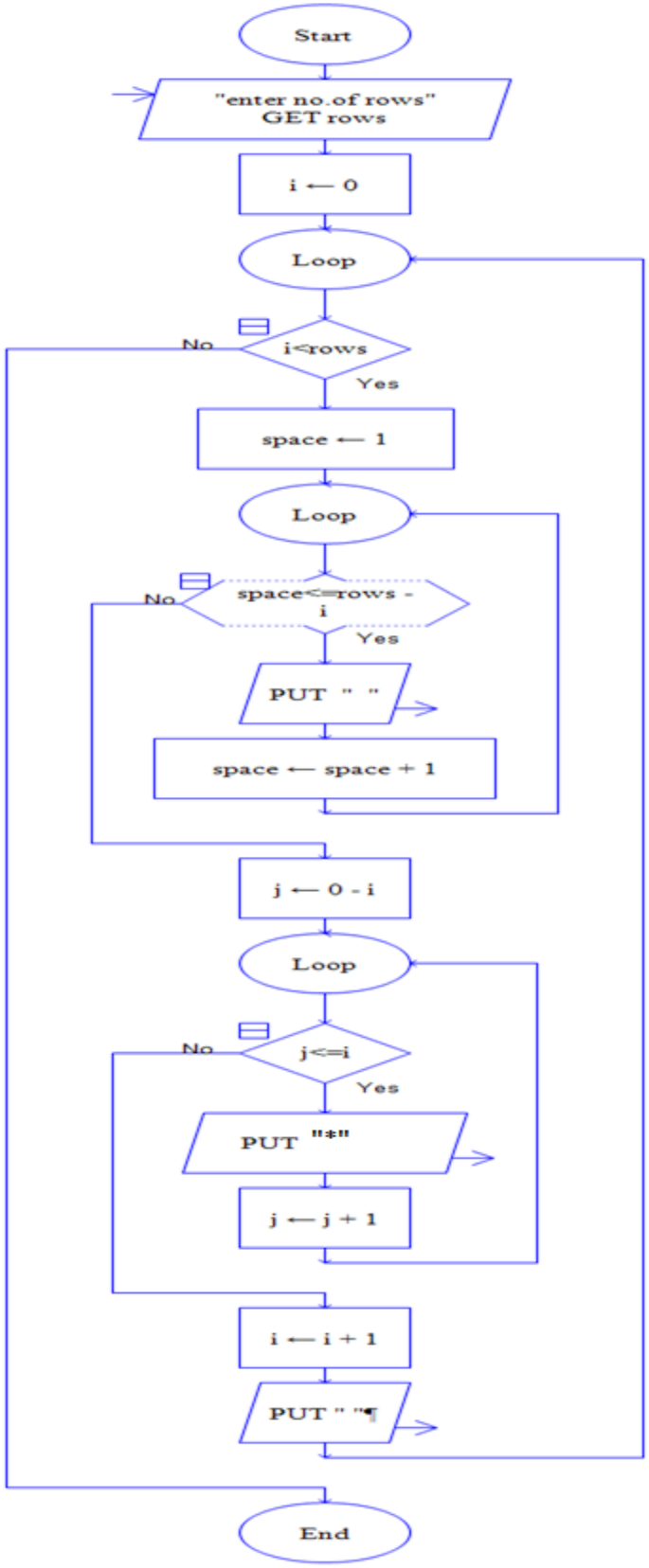
Step 3 :Print star

Use the print() function in each iteration of nested for loop to display the symbol (like a star (asterisk *)).

Step 4 : Add new line after each iteration of outer loop

Add a new line after each iteration of the outer loop so that the pattern display appropriately

FlowChart :


Start
"enter no.of rows"
GET rows
i ← 0
Loop
No
i<rows
Yes
space ← 1
Loop
No
space<=rows - i
Yes
PUT " "
space ← space + 1
j ← 0 - i
Loop
No
j<=i
Yes
PUT "*"
j ← j + 1
i ← i + 1
PUT " "¶
End

PROGRAM :

```
#include <stdio.h>
int main() {
  int i, space, rows, k = 0;
  printf("Enter the number of rows: ");
  scanf("%d", &rows);
  for (i = 1; i <= rows; ++i, k = 0) {
    for (space = 1; space <= rows - i; ++space) {
      printf("  ");
    }
    while (k != 2 * i - 1) {
      printf("* ");
      ++k;
    }
    printf("\n");
  }
  return 0;
}
```

OUTPUT :

Enter the number of rows : 5

```
        *
      * * *
    * * * * *
  * * * * * * *
* * * * * * * * *
```

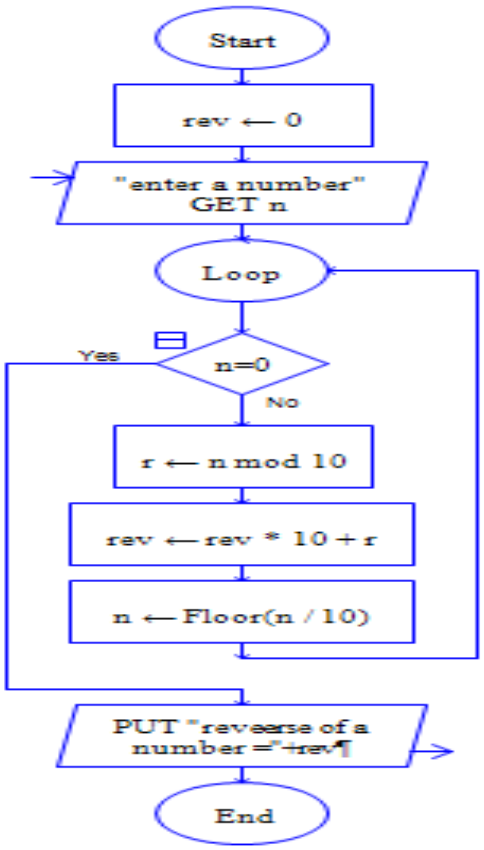
MODULE 6 – PROGRAM 6.6

AIM : Write a C Program to find reverse of a given number.

ALGORITHM :

Step-1: Start
Step-2: Read n
Step-3: rev=0
Step-3: repeat step 4 to 6 until n!=0
Step-4: Calculate r=n%10
Step-5: rev=rev*10+r
Step-6: Calculate n=n/10
Step-7: display rev
Step-7: Stop

FLOWCHART:



PROGRAM :

```
#include<stdio.h>
int main()
{
  int n,r,rev=0;
  printf("enter the number \t");
  scanf("%d",&n);
  printf("\n the reverse of the given no is \t");
    while(n>0)
  {
   r = n%10;
   rev=rev*10+r;
   n = n/10;
  }
printf("%d",rev);
return 0;
}
```

OUTPUT :

enter the number 145

the reverse of the given no is 541

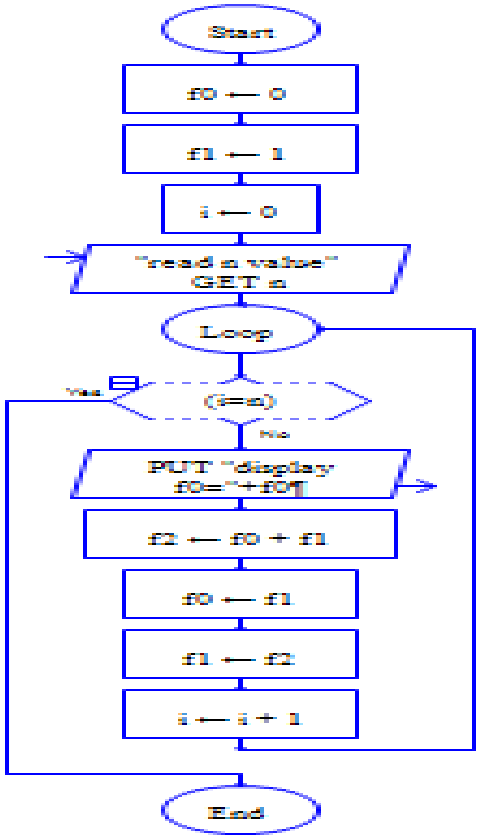
MODULE 6 - Program 6.7

AIM : write a C program to generate Fibonacci series unto n terms.

ALGORITHM :

step 1 : start
step 2 : let f0=0,f1=1
step 3 : read n
step 4 : repeat step 5 to step 8 'n' number of times
step 5 : display f0
step 6 : f2=f0+f1
step 7 : f0=f1
step 8 : f1=f2
step 9 : stop

FLOWCHART :



PROGRAM 5.1 :

```
#include<stdio.h>
 #include<conio.h>
 int main()
 {
  int n,f1=0,f2=1,f,i=1;
  printf("\n enter the number of fibonacci numbers \t");
  scanf("%d",&n);
  printf("\n fibonacci series is \n\n");
  while(i<=n)
  {
   f=f1+f2;
   printf("%d\t",f1);
   f1=f2;
   f2=f;
   i++;
  }
  return 0;
 }
```

OUTPUT :

```
enter the number of fibonacci numbers  8

 fibonacci series is

0       1       1       2       3       5       8       13
```

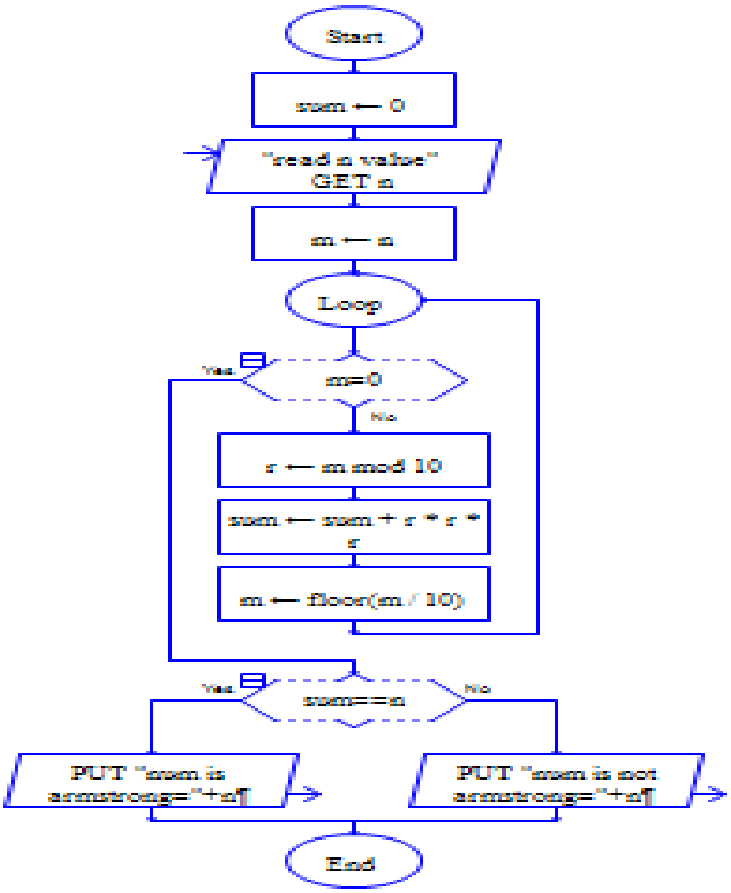
MODULE 6 – PROGRAM 6.8

AIM : Write a C Program to find the given number is Armstrong or not

ALGORITHM :

Step 1 : start
Step 2 : read the number n [initialize m=n]
Step3 : repeat step3 through 5 until(m!=0)
Step4 : calculate r=m%10
Step5 : sum=sum+r*r*r
m=m/10
step6 : if (sum ==n)
display it is Armstrong number
otherwise
display it is not a Armstrong number
step8 : stop

FLOWCHART:



PROGRAM :

```
#include<stdio.h>
int main()
 {
  int n,r,s=0,m;
    printf("\n enter the number \t");
  scanf("%d",&n);
  m=n;
  while(n!=0)
  {
    r=n%10;
    s=s+r*r*r;
    n=n/10;
  }
  if(s==m)
     printf("\n it is armstrong number \n");
  else
     printf("\n it is not an armstrong number \n");
return 0;
}
```

OUTPUT :

```
enter the number       153
 it is armstrong number

enter the number       678
 it is not an armstrong number
```

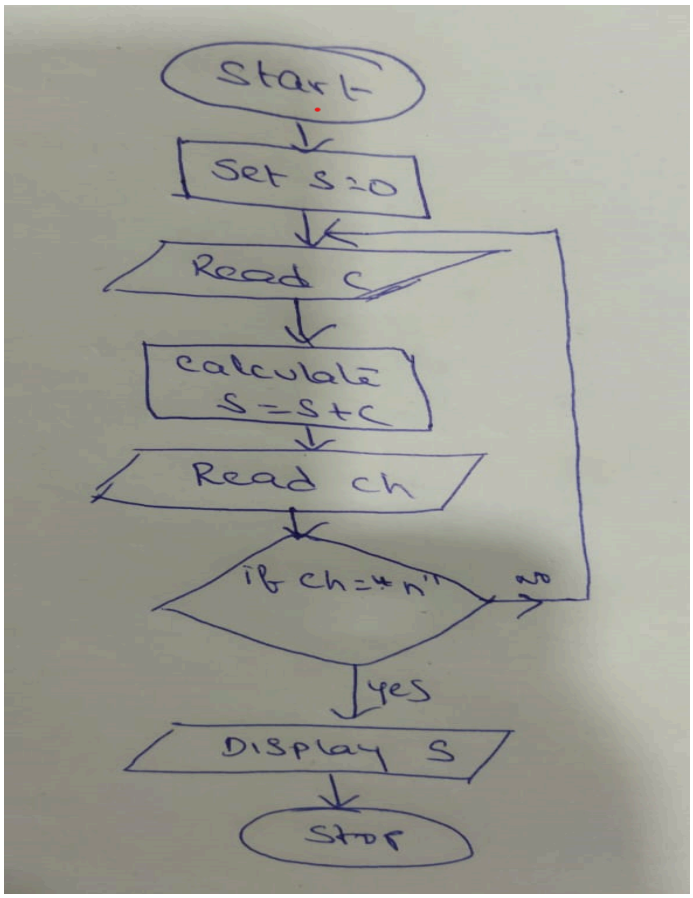
MODULE 6 – PROGRAM 6.9

AIM : Write a C Program to read the prices of items as float one by one until they decide to stop. Use a loop to continuously prompt the user for item prices and calculate the total cost. Once the user decides to stop, print the total cost as float up to two decimal point

ALGORITHM :

Step 1 : start
Step 2: set ch=”y” and i=1, s=0
Step 3 : repeat step 4 through 6 until(ch!=”n”)
Step 4: read the cost of item c
Step 5 : calculate s=s+c
Step 6 : read ch
Step 7 : display total cost s.
step8 : stop

FLOWCHART:



PROGRAM:

```
#include<stdio.h>
#include<string.h>
int main()
{
  int i=1;
   double c,s=0;
   char ch[2];
   while(1)
   {
   printf("\n Enter the price of item%d : ",i);
   scanf("%lf",&c);
   s=s+c;
   i=i+1;
   printf("\n do you want continue (y/n) : ");
   scanf("%s",ch);
   if(strcmp(ch,"n")==0)
     break;
   }
  printf("\n Total cost = %.2lf",s);
   return 0;
}
```

OUTPUT :

```
Enter the price of item1 : 50

do you want continue (y/n) : y

Enter the price of item2 : 25

do you want continue (y/n) : y

Enter the price of item3 : 45

do you want continue (y/n) : n

Total cost = 120.00
```

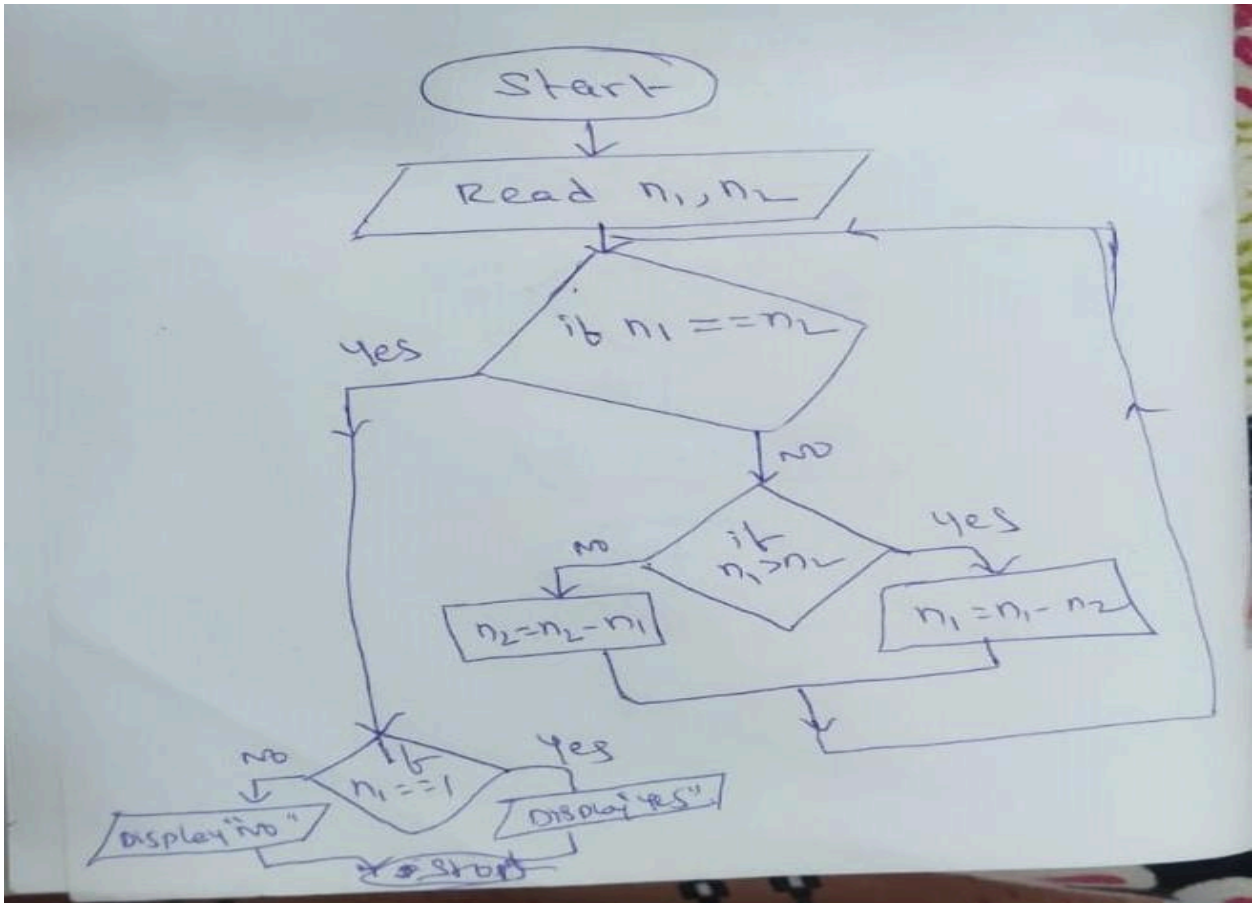
MODULE 6 – PROGRAM 6.10

AIM : Write a C Program to Find two numbers are co-primes or not. Two integers are co-prime if the only common divisor between the two integers is one.

ALGORITHM :

Step 1 : start
Step 2: read n1,n2
Step 3 : repeat step 4 until(n1==n2)
Step 4: if n1>n2 do step 5 otherwise do step 6
Step 5 : n1=n1-n2
Step 6 : n2=n2-n1
Step 7 : if n1==1
display co-prime
otherwise
display not co-prime
step8 : stop

FLOWCHART :



PROGRAM :

```
#include<stdio.h>
int main()
{
   int n1, n2;
   printf("\n Enter any two numbers : ");
   scanf("%d %d",&n1,&n2);
   printf("%d and %d are ",n1,n2);
   while(n1!=n2)
   {
   if(n1>n2)
     n1=n1-n2;
   else
     n2=n2-n1;
   }
```

```
    if(n1==1)
      printf("Co-primes");
    else
      printf("Not Co-primes");
    return 0;
}
```

OUTPUT :

```
 Enter any two numbers : 12 19
12 and 19 are Co-primes
```

```
 Enter any two numbers : 6 16
6 and 16 are Not Co-primes
```