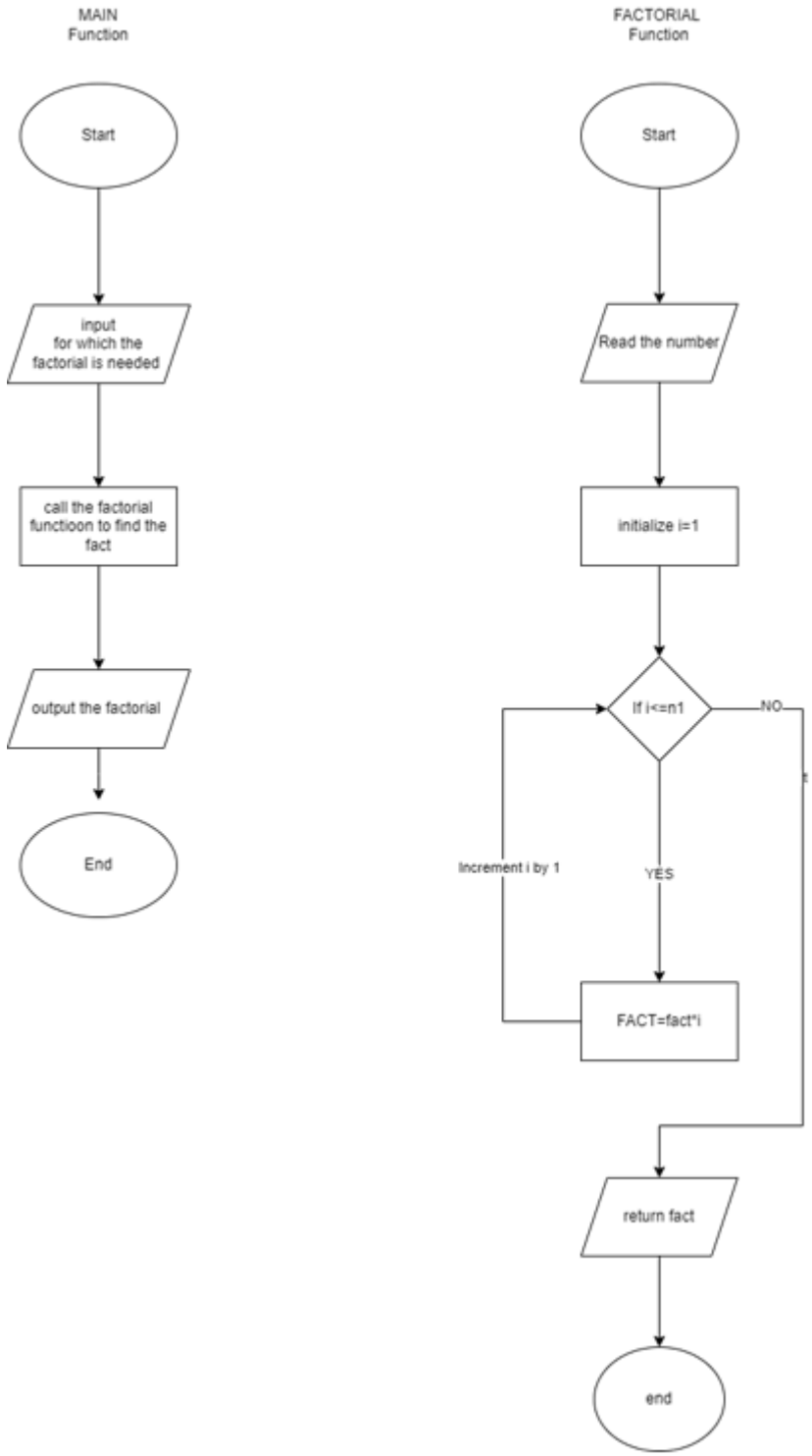
Name	Debjit Ghosal
UID no.	2023300065
Experiment No.	3

AIM:	**Apply the concept of functions to incorporate modularity**
Program 1	
PROBLEM STATEMENT :	Write a function to find the factorial of a number
ALGORITHM:	Factorial function Step 1: Start Step2: enter the 1 numbers Step3: create a for loop with f=1; condition f<=num;f++ Step4 : prod=prod*f Step5 : return the value of prod Step6 :end Main function Step 1: Start Step2: enter the 1 numbers Step3: call the factorial functioon to find the fact Step4: print the fact
FLOWCHART:	


MAIN
Function
Start
input
for which the
factorial is needed
call the factorial
functioon to find the
fact
output the factorial
End
FACTORIAL
Function
Start
Read the number
initialize i=1
If i<=n1
NO
YES
Increment i by 1
FACT=fact*i
return fact
end

PROGRAM:

```
#include<stdio.h>

int factorial(int num)
{
int prod=1;
for(int f=1;f<=num;f++)
prod=prod*f;
return prod;
}
int main()
{
  int n;
printf("enter n :");
scanf("%d",&n);
int fact;
fact=factorial(n);
printf("factorial is %d",fact);

return 0;
}
```

RESULT:

```
Enter n:5

The factorial of 5 is : 120
```

Program 2

PROBLEM STATEMENT :

Write a function which takes a range as input. Print all the numbers in the range with '*' in front of prime numbers only.

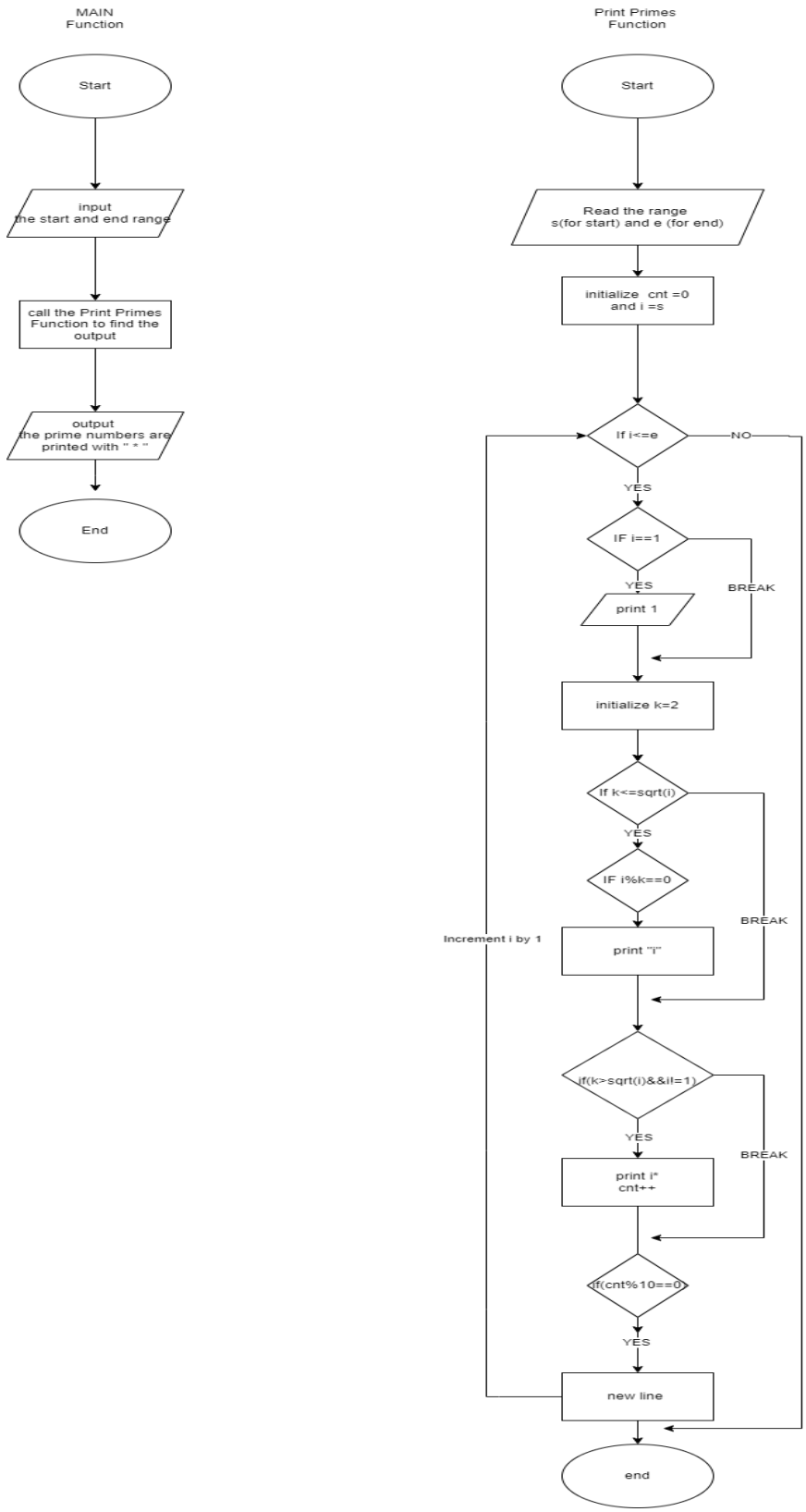
ALGORITHM:

Main function
Step 1: Start
Step2: enter the start and end range
Step3: print the prime numbers with a star using the function call
Step4: end

Void Main
Step1: start
Step2: initialize counter to 0
Step3: crate a for (from step 4 to step 9) loop with I = start of the range til

	condition i<=end of the range. Step 4 : intialize k Step 5: if i==k Then print “1” Step 6: create a for loop(from 7 to 9) by initializing K=2 ; till k<=sqrt(i) Step 7: if k%i==0 Then print “i” Step8: if k<sqrt(i) && i!=1 Then print “i*” And increase counter by 1 Step9: if cnt%10 , then go to new line Step 10: end

FLOWCHART:


MAIN Function
Start
input the start and end range
call the Print Primes Function to find the output
output the prime numbers are printed with " * "
End
Print Primes Function
Start
Read the range s(for start) and e (for end)
initialize cnt =0 and i =s
If i<=e
NO
YES
IF i==1
YES
BREAK
print 1
initialize k=2
If k<=sqrt(i)
YES
IF i%k==0
BREAK
Increment i by 1
print "i"
if(k>sqrt(i)&&if=1)
YES
BREAK
print i" cnt++
if(cnt%10==0)
YES
new line
end

PROGRAM:	

```
#include<stdio.h>
#include<math.h>
void print_primes(int,int);
int main()
{
   int s,e;
   printf("Enter the start and end points :");
   scanf("%d %d",&s,&e);
   print_primes(s,e);//function call
   return 0;
}
//function definition
void print_primes(int start,int end)
{
    int cnt=0;
    for(int i=start;i<=end;i++)
    {
       int j;
       if(i==1)
          printf("1 ");
        for(j=2;j<=sqrt(i);j++)//prime-check
       if(i%j==0)// to get no remainder
         {
         printf("%d ",i);
         break;
         }
        if(j>sqrt(i)&&i!=1)
          printf("%d*",i);
        cnt++;
        if(cnt%10==0)
        {
         printf("\n");

        }
   }//outer for
}
```

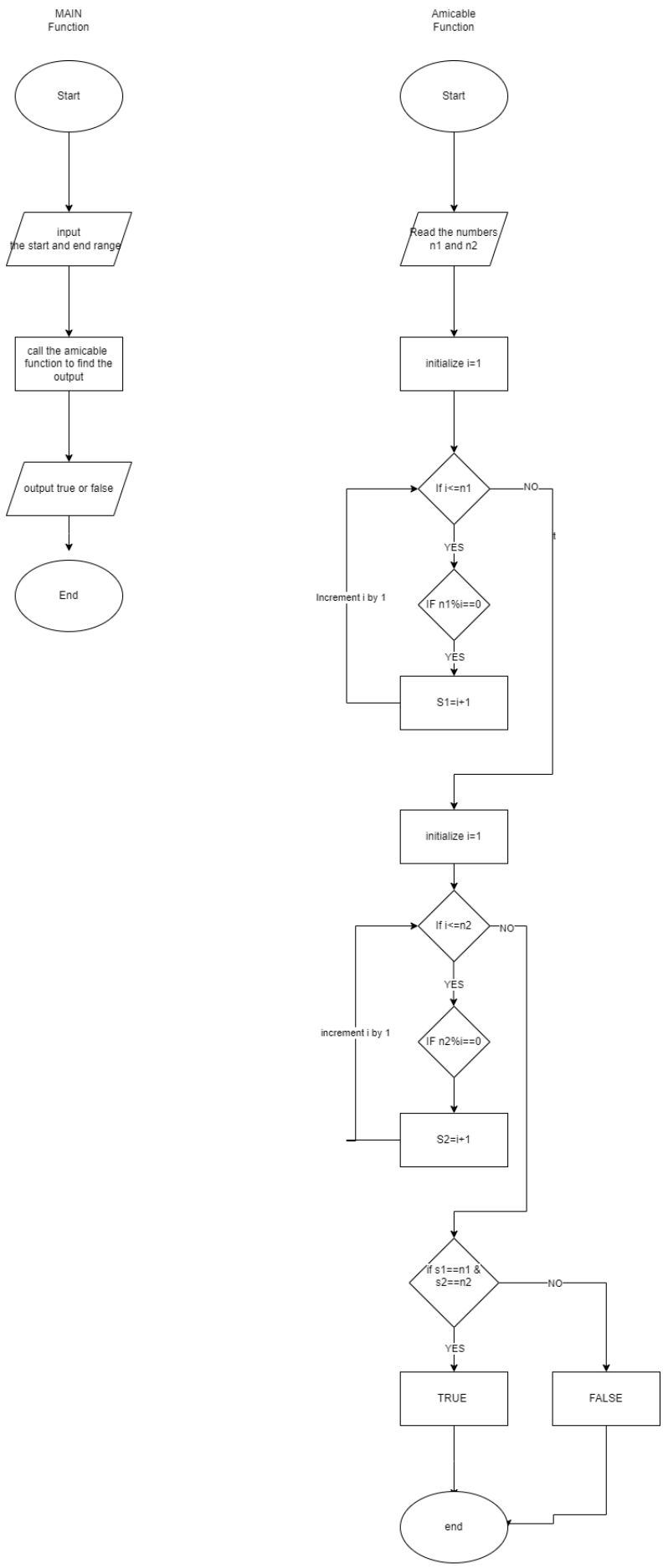
RESULT:

```
psipl@psipl-OptiPlex-3000: ~/Desktop/2023300065
psipl@psipl-OptiPlex-3000:~$ cd Desktop/
psipl@psipl-OptiPlex-3000:~/Desktop$ cd 2023300065
psipl@psipl-OptiPlex-3000:~/Desktop/2023300065$ gcc print_primes.c -lm
psipl@psipl-OptiPlex-3000:~/Desktop/2023300065$ ./a.out
Enter the start and end points :1 100
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 56 57 58 59*60
61*62 63 64 65 66 67*68 69 70
71*72 73*74 75 76 77 78 79*80
81 82 83*84 85 86 87 88 89*90
91 92 93 94 95 96 97*98 99 100
psipl@psipl-OptiPlex-3000:~/Desktop/2023300065$
```

Program 3	
PROBLEM STATEMENT:	Write a function which takes as parameters two positive integers and returns TRUE if the numbers are amicable and FALSE otherwise. A pair of numbers is said to be amicable if the sum of divisors of each of the numbers (excluding the no. itself) is equal to the other number. Ex. 1184 and 1210 are amicable
ALGORITHM:	Step 1: Start Step2: enter the 2 numbers Step3: check if the numbers are amicable are not but calling upon the amicable function(that is user defined) Step 4: if the function amicable return 1 then print true Or else print false Step 5:end Amicable function Step1:input n1 and n2 and create 2 variables s1 and s0 Step 2:creat an for loop with i=1; condition i<n1 ; increment I by 1 after each loop Step3: if n1%i=0

	Then s1=i+1 Step4:create a for loop with I=1;condition I<n2; and increment I by 1 Step 5: if s2%i=o Then s2=1+i Step6: if s1=n1 and s2=n2 Then return 1 Else return 0 Step 7:end

FLOWCHART:

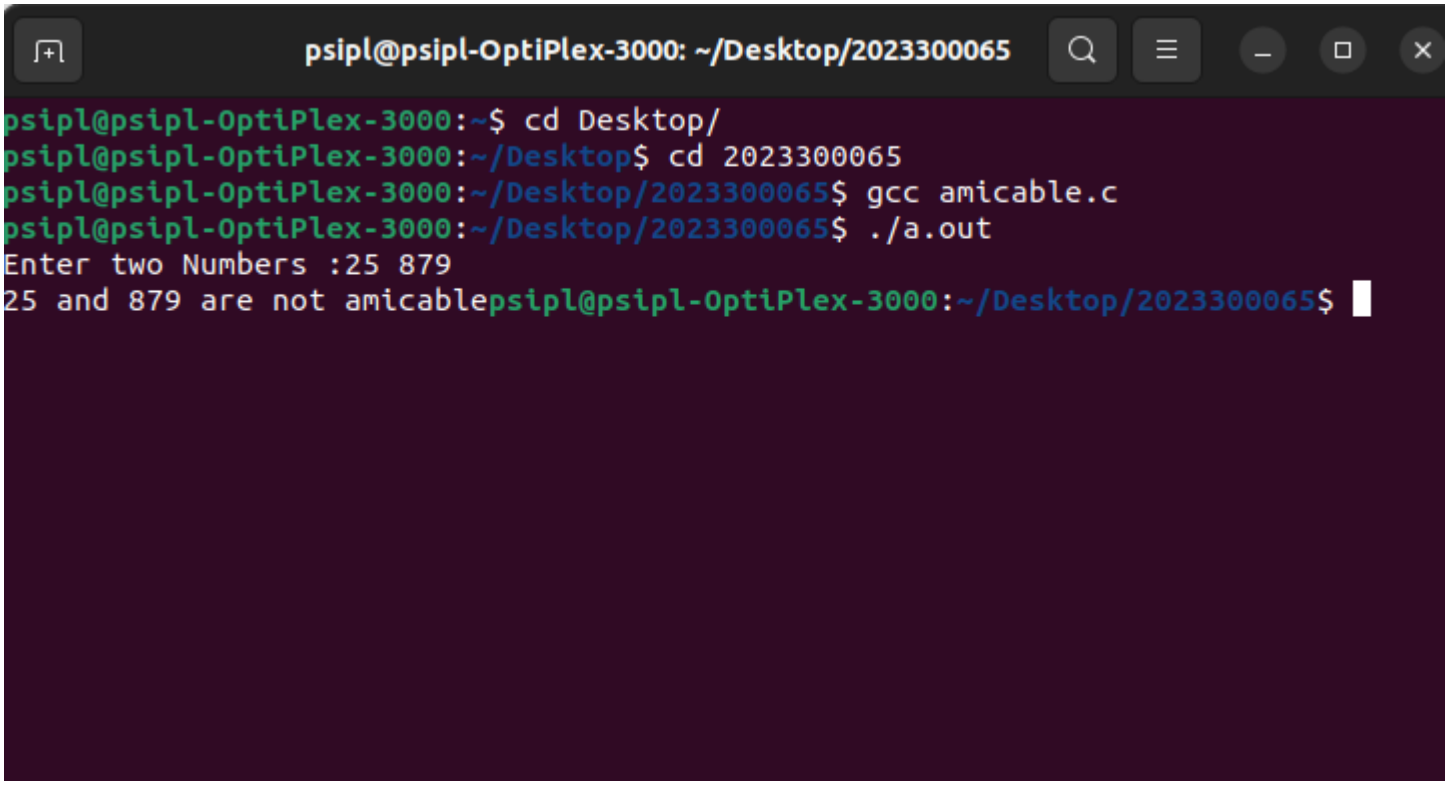

MAIN Function
Start
input the start and end range
call the amicable function to find the output
output true or false
End
Amicable Function
Start
Read the numbers n1 and n2
initialize i=1
If i<=n1
NO
YES
Increment i by 1
IF n1%i==0
YES
S1=i+1
initialize i=1
If i<=n2
NO
YES
increment i by 1
IF n2%i==0
S2=i+1
if s1==n1 & s2==n2
NO
YES
TRUE
FALSE
end

PROGRAM:	

```
#include<stdio.h>
int amicable(int,int);
int main()
{
   int num1,num2;
   printf("Enter two Numbers :");
   scanf("%d %d",&num1,&num2);
   int k=amicable(num1,num2);//function cal
   if(k==1)
      printf("%d and %d are amicable",num1,num2);
   else
            printf("%d and %d are not amicable",num1,num2);
            return 0;

}
int amicable(int n1,int n2)
{
   int s1=0,s2=0;
   //factors of n1
   for(int i=1;i<n1;i++)
     if(n1%i==0)
         s1+=i;
   for(int i=1;i<n2;i++)
     if(n2%i==0)
         s2+=i;
     if(s1==n2 && s2==n1)
     return 1;
  else
      return 0;
}
```

RESULT:



CONCLUSION:	By this experiment I have learnt various functions that we can achieve using C program and using other library like the math.io used for mathematic operations.