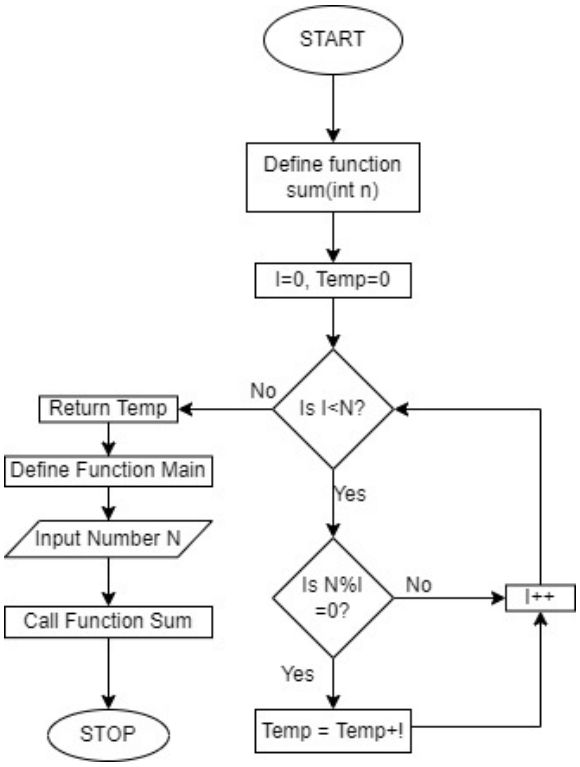
Name	Vineet Parmar
UID no.	2021300092
Experiment No.	3

AIM:	Apply the concept of functions to incorporate modularity
Program 1	
PROBLEM STATEMENT :	Write a function to find the sum of the proper divisors of a given number 'n'. The proper divisors of a number 'n' are the numbers less than n that divide it; they do not include n itself
ALGORITHM:	1. START 2. Define function sum with an integer parameter N 3. I=1, Temp=0 4. If N%I = 0 Temp = Temp+I 5. I++ 6. Repeat 4, and 5 till I<N 7. Return Temp 8. Define function main 9. Input number N 10. Call Function Sum 11. STOP

FLOWCHART:



PROGRAM:

```
#include<stdio.h>
#include<conio.h>
int sum(int n)
{
   int temp=0;
   for(int i=1;i<n;i++)
   {
      if(n%i==0)
      {
         temp+=i;
      }
   }
   return temp;
}
int main()
{
   int i;
   printf("Enter the number: ");
   scanf("%d",&i);
   printf("Sum of all divisors: %d",sum(i));
   return 0;
}
```

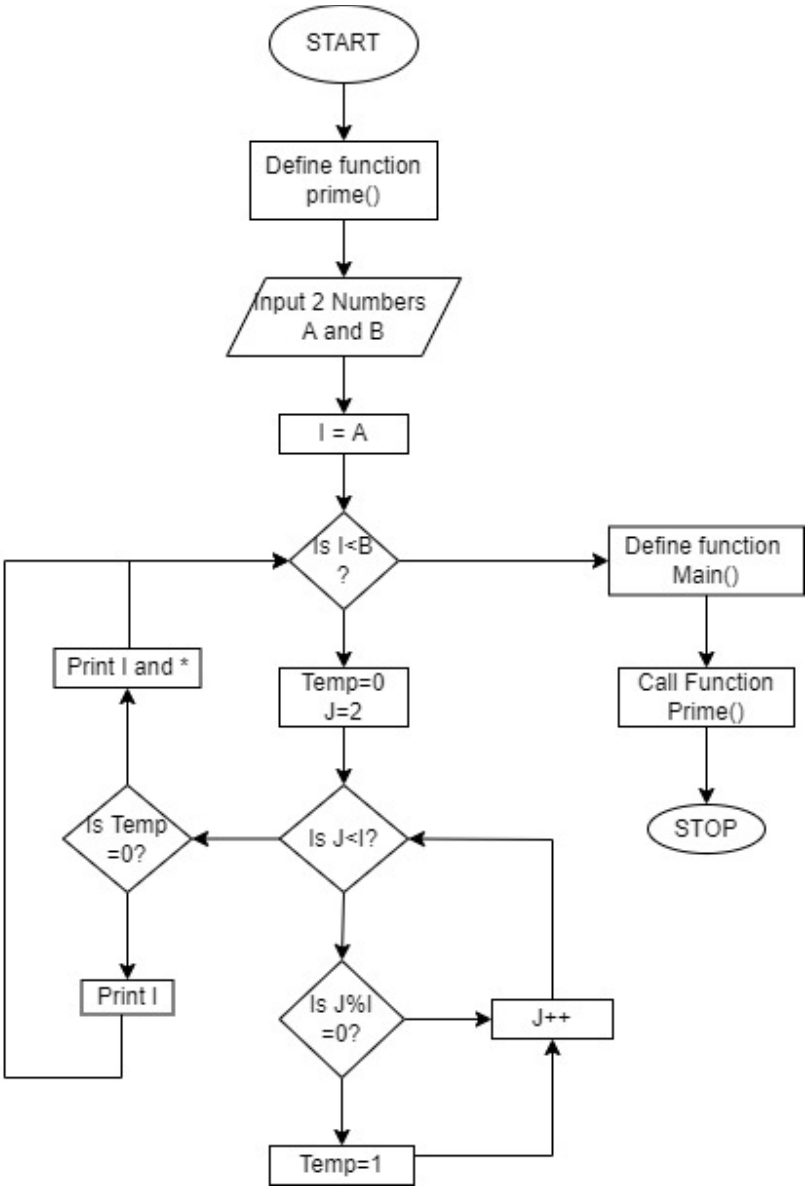
RESULT:

```
Enter the number: 12
Sum of all divisors: 16

...Program finished with exit code 0
Press ENTER to exit console.
```

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:	1. START 2. Define function prime. 3. Input 2 numbers A and B 4. I=A 5. Temp=0, J=2 6. If J%I=0 Temp = 1 7. J++ 8. Repeat 6, 7 till J<I 9. If Temp=1 Print Number Else If Temp=0 Print Number and * 10. If (I-A+1)%10=0 Print New Line 11. I++ 12. Repeat steps 5 to 11 till I=B 13. Define function Main 14. Call function Prime 15. STOP

FLOWCHART:



PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void prime()
{
   printf("Enter two numbers: ");
   int a,b,temp=0;
   scanf("%d",&a);
   scanf("%d",&b);
   for(int i=a;i<=b;i++)
   {
      temp=0;
      for(int j=2;j<i;j++)
```

```
        {
            if(i%j==0)
            {
                temp=1;
            }
        }
        if(temp==1)
        {
            printf("%d ",i);
        }
        else
        {
            printf("%d* ",i);
        }
        if((i-a+1)%10==0)
        {
            printf("\n");
        }
    }
}
int main()
{
    prime();
    return 0;
}
```

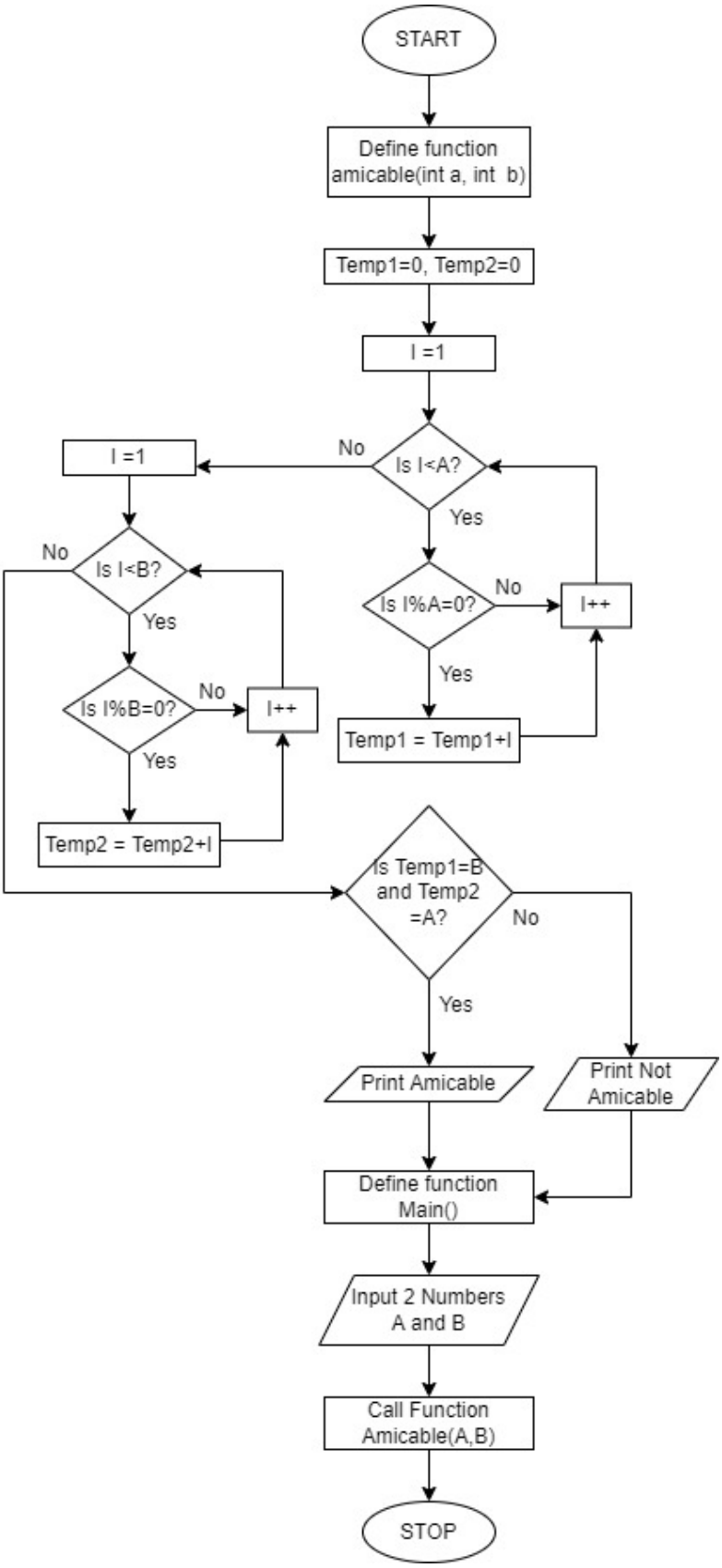
RESULT:

```
Enter two numbers: 3
37
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*

...Program finished with exit code 0
Press ENTER to exit console.
```

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:	1. START 2. Define function amicable with two integer parameters A,B 3. Temp1 =0 , Temp2 =0 4. I=1 5. If A%I=0 Temp1 = Temp1+I 6. I++ 7. Repeat 4,5 and 6 Till I<A 8. I=1 9. If B%I =0 Temp2 = Temp2 + I 10. I++ 11. Repeat 8,9 and 10 Till I<B 12. If Temp1=B and Temp2=A Print Amicable Else Print Not Amicable 13. Define Function Main 14. Input 2 Numbers A, B 15. Call Function Amicable 16. STOP

FLOWCHART:


START
Define function amicable(int a, int b)
Temp1=0, Temp2=0
I =1
Is I<A?
No
Yes
I =1
Is I%A=0?
No
I++
Yes
Temp1 = Temp1+I
Is I<B?
No
Yes
Is I%B=0?
No
I++
Yes
Temp2 = Temp2+I
Is Temp1=B and Temp2 =A?
No
Yes
Print Amicable
Print Not Amicable
Define function Main()
Input 2 Numbers A and B
Call Function Amicable(A,B)
STOP

PROGRAM:	

```
#include<stdio.h>
#include<conio.h>
int amicable(int a,int b)
{
   int temp1=0,temp2=0;
   for(int i=1;i<a;i++)
   {
      if(a%i==0)
      {
         temp1+=i;
      }
   }
   for(int i=1;i<b;i++)
   {
      if(b%i==0)
      {
         temp2+=i;
      }
   }
   if(temp1==b&&temp2==a)
   {
      return 1;
   }
   else
   {
      return 0;
   }
}
int main()
{
   int a,b;
   printf("Enter 2 numbers: ");
   scanf("%d",&a);
   scanf("%d",&b);
   if(amicable(a,b)==1)
   {
      printf("TRUE");
   }
   else if(amicable(a,b)==0)
   {
      printf("FALSE");
   }
```

```
    return 0;
}
```

RESULT:

```
Enter 2 numbers: 1184
1210
TRUE

...Program finished with exit code 0
Press ENTER to exit console.
```

Program 4

PROBLEM STATEMENT: Write a function to find out whether given numbers are relatively prime or not. A number is relatively prime if the '1' is the only common factor between the two numbers.

ALGORITHM:

1. START
2. Define Function Rel_Prime with two integer parameters A,B
3. Define 2 Arrays P, Q of length 100 each,J=0,K=0,Temp=0
4. I=1
5. If A%I=0
   P[J]=I
   J++
6. I++
7. Repeat 5 and 6 till I<A
8. I=1
9. If B%I=0
   Q[K]=I
   K++
10. I++
11. Repeat 9 and 10 till I<B
12. I=1
13. If A%Q[I]=0
    Temp=1
14. I++
15. Repeat 13 and 14 till I<K
16. If Temp=1
    Print Not Relatively Prime
    Else If Temp=0
    Print Relatively Prime
17. Define Function Main
18. Input 2 numbers in Ascending Order
19. Call Function Rel_Prime
20. STOP

PROGRAM:	

```
#include<stdio.h>
#include<conio.h>
int rel_prime(int a, int b)
{
   int p[100],q[100],j=0,k=0,temp=0;
   for(int i=1;i<a;i++)
   {
      if(a%i==0)
      {
         p[j]=i;
         j++;
      }
   }
   for(int i=1;i<a;i++)
   {
      if(b%i==0)
      {
         q[k]=i;
         k++;
      }
   }
   for(int i=1;i<k;i++)
   {
      if(a%q[i]==0)
      {
         temp=1;
      }
   }
   if(temp==1)
   {
      return 0;
   }
   else
   {
      return 1;
   }
}
int main()
{
   int a,b;
   printf("Enter two numbers, the smaller one first: ");
   scanf("%d",&a);
```

```
    scanf("%d",&b);
    if(rel_prime(a,b)==1)
    {
        printf("Numbers are relatively prime");
    }
    else if(rel_prime(a,b)==0)
    {
        printf("Numbers are not relatively prime");
    }
    return 0;
}
```

RESULT:

```
Enter two numbers, the smaller one first: 8
9
Numbers are relatively prime

...Program finished with exit code 0
Press ENTER to exit console.
```

CONCLUSION: In this session, we learned more about functions and the parameters that accompany them. We also learned about different return types such as int and void and also we learned how to call them in the main function.