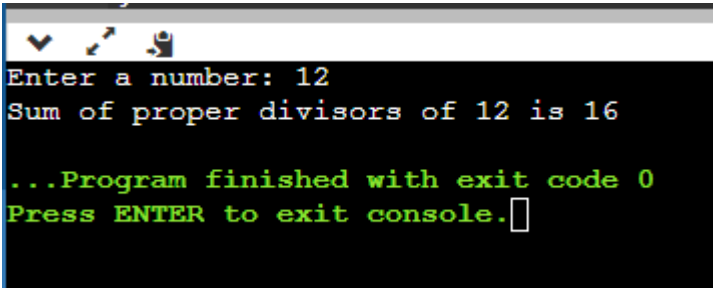


Name	Adwait S Purao
UID no.	2021300101
Experiment No.	3

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 divsum with integer parameter n 3.i,sum=0 4.for(i=1;i,n;i++) 5.if(n%i==0) 6.sum+=i 7.return sum 8.In main function input number n 9.Call intdivsum 10.STOP
FLOWCHART:	

--	--

PROGRAM:	<pre> #include <stdio.h> int divSum(int n) { int sum=0, i; for(i=1; i<n; i++) { if(n%i == 0) { sum += i; } } return sum; } int main() { int n; printf("Enter a number: "); scanf("%d", &n); printf("Sum of proper divisors of %d is %d", n, divSum(n)); return 0; } </pre>
-----------------	---

RESULT:	
----------------	--

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 void prime which takes input n1 and n2 3. for(i=n1; i<=n2; i++) 4. for(j=2; j<i; j++) .if(i%j==0) prmchk=0 break 5. if(prmchk == 1) print * 6.else print i 7.Define function main 8.Input two numbers 9.Call function prime 10.STOP
FLOWCHART:	
PROGRAM:	<pre>#include <stdio.h> void prime(int n1, int n2) { int i, j, prmchk=1;</pre>

```
for(i=n1; i<=n2; i++)
{
    for(j=2; j<i; j++)
    {
        if(i%j == 0){
            prmchk = 0;
            break;
        }
    }
    if(prmchk == 1){
        printf("%d* ", i);
    }else{
        printf("%d ", i);
    }
    prmchk = 1;
}
}
int main()
{
    int n1, n2;
    printf("Enter first number: ");
    scanf("%d", &n1);
    printf("Enter last number: ");
    scanf("%d", &n2);
    prime(n1, n2);
    return 0;
}
```

--	--

RESULT:

```

input
Enter first number: 1
Enter last number: 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 3
9 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

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

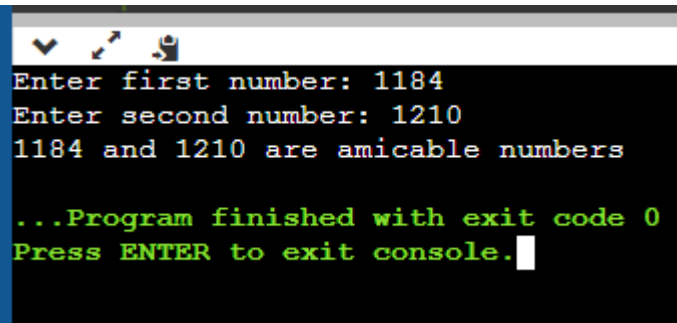
```

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 inputs n1 and n2 3.sum1=0, sum2=0, i 4. for(i=1; i<n1; i++) 5. if(n1%i == 0) .sum1 +=i 6.for(i=1; i<n2; i++) 7.if(n2%i == 0) sum2 += i 8. if(sum1 == n2 && sum2 == n1) return 1 else return 0 9..Define function main 10.Input two numbers 11.Call function function amicable 12.STOP
FLOWCHART:	

PROGRAM:	<pre>#include <stdio.h> int amicable(int n1, int n2) { int sum1=0, sum2=0, i; for(i=1; i<n1; i++) { if(n1%i == 0) { sum1 += i; } } for(i=1; i<n2; i++) { if(n2%i == 0) { sum2 += i; } } if(sum1 == n2 && sum2 == n1) { return 1; } else{ return 0; } } int main() { int n1, n2, tf; printf("Enter first number: "); scanf("%d", &n1); printf("Enter second number: "); scanf("%d", &n2); tf = amicable(n1, n2); if(tf == 1){ printf("%d and %d are amicable numbers", n1, n2); } }</pre>

	<pre> }else{ printf("%d and %d are not amicable numbers", n1, n2); } return 0; } </pre>
--	---



```

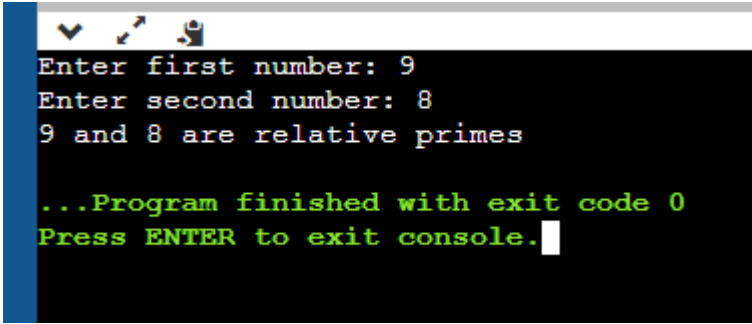
Enter first number: 1184
Enter second number: 1210
1184 and 1210 are amicable numbers
...Program finished with exit code 0
Press ENTER to exit console.

```

RESULT:

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:	<ol style="list-style-type: none"> 1.START 2.Define function relprime,with two integer parameters n1 and n2 3. for(i=2; i<n1 i<n2; i++) <ul style="list-style-type: none"> if(n1%i == 0 && n2%i == 0) Print n1 and n2 are not relative primes rel=0 break 4. if(rel == 1) <ul style="list-style-type: none"> Print n1 and n2 are relative primes 5.Define function main and input two variables 6.call function relprime 7.STOP

FLOWCHART:	
PROGRAM:	<pre>#include <stdio.h> void relPrime(int n1, int n2) { int i, rel=1; for(i=2; i<n1 i<n2; i++) { if(n1%i == 0 && n2%i == 0) { printf("%d and %d are not relative primes", n1, n2); rel = 0; break; } } if(rel == 1) { printf("%d and %d are relative primes", n1, n2); } } int main() { int n1, n2; printf("Enter first number: "); scanf("%d", &n1); printf("Enter second number: "); scanf("%d", &n2); relPrime(n1, n2); return 0; }</pre>

RESULT:	 <pre> Enter first number: 9 Enter second number: 8 9 and 8 are relative primes ...Program finished with exit code 0 Press ENTER to exit console. </pre>
Program 5	
PROBLEM STATEMENT:	
ALGORITHM:	
FLOWCHART:	
PROGRAM:	

RESULT:	
CONCLUSION:	