DSA LAB PROGRAMS -Functions:

By Sindhu Sankati AP19110010477 CSE-G

PROGRAM 22

Title: Sum of natural numbers

Objective:

Write a C Program to find the sum of natural numbers using function.

Explanation:

Given n where n>=0, we have sum = 1+2+....+n

Pseudo Code:

>Sum N natural numbers

```
1. SET = n
2. sum = 0
3. FOR i = 1 till n DO
   {
4.     sum += i
   }
5. RETURN sum
6. END
```

Code:

```c
#include <stdio.h>
//function to add from 1 to n
int sum_natural(int n){
     int i,sum=0;
     //loops runs from 1 to n
     for(i=1;i<=n;i++)
          sum += i;
     return sum;
}
int main(int argc, char const *argv[]){
     int n;
     //inputs n
     printf("Enter some n\n");
     scanf("%d",&n);
     //prints sum
     printf("Sum till of %d natural numbers is
%d",n,sum_natural(n));
```

```
        return 0;
    }
```

Outputs:

```
sravImac in ~/Desktop/sindhu/Dsa-lab/functions
$ ./a.out
Enter some n
7
Sum till of 7 natural numbers is 28
```

Conclusion:

This code is giving expected result. The time complexity is O(n). But it can be improved to O(1) ,by using the maths formula sum_n = n*(n+1)/2.

PROGRAM 23

Title: Factorial

Objective:

Write a C Program to find factorial of number using recursion.

Explanation:

Given a number n, where n>=0, factorial of number n! = 1x2x3....(n-1)x(n)

Pseudo Code:

>Factorial

```
1. RECURSION(n)
   {
2.    IF n==0 or n ==1 : RETURN 1
3.    RETURN n * RECURSION(n-1)
   }
4. END
```

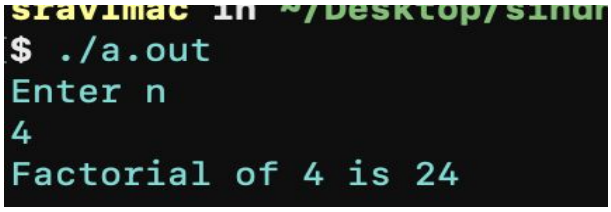
Code:

```c
#include <stdio.h>

int recurFact(int n){
    //stop condion for recursive stack
    if(n == 0|| n==1)
        return 1;
    return n*recurFact(n-1);
}
```

```c
int main(int argc, char const *argv[])
{
    int n;
    printf("Enter n\n");
    scanf("%d",&n);
    printf("Factorial of %d is %d\n",n,recurFact(n));
    return 0;
}
```

Outputs:



Conclusion:

This code is giving expected result. The time complexity is O(n*), including the time for calling the function.

PROGRAM 23

Title: Fibonacci

Objective:

Write a C Program to generate the Fibonacci series.

Explanation:

For finding the nth fibonocci number, Fn = Fn-1 + Fn-2.

Pseudo Code:

>Fibonacci

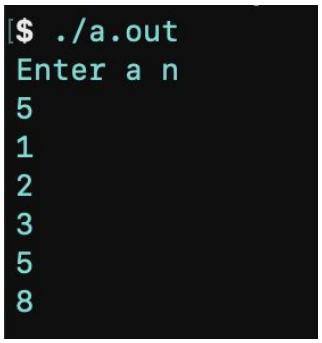
```
1. a=0,b=1,c=0
2. FOR i btw 0 to n:
   {
3.     c=a+b, a=b, b=c
4.     PRINT c
   }
5. END
```

Code:

```c
#include <stdio.h>
```

```c
//prints the fibanocci series till n numbers
void fibonnaciSeries(int n){
     int i, a = 0, b=1,c;
     for(i=0;i<n;i++){
          c = a+b;
          a = b;
          b = c;
          //prints ith fibanoccci number
          printf("%d\n",c);
     }
}
int main(int argc, char const *argv[])
{
     int n;
     //inputting n
     printf("Enter a n\n");
     scanf("%d",&n);
     //calling function to print the first n fibanooci numbers.
     fibonnaciSeries(n);
     return 0;
}
```

Outputs:



Conclusion:

This code is giving expected result. The time complexity is O(n).