FUNCTIONS

By ASRITHA MEKA, AP19110010224,CSE-G

22)

TITLE:

SUM OF NATURAL NUMBERS

OBJECTIVE:

Write a c program to find the sum of natural numbers using function.

EXPLANATION:

n>=0, we have the sum =1+2+3+4+5+....+n

PSEUDO CODE:

```
1. set=n
2. s=0
3. For x=1 till n
   {
4.     s+=x
   }
5. Return s
6. END
```

CODE:

```
#include <stdio.h>

void main()
{
        int X, num, s = 0;

        printf("Enter the integer \n");
        scanf ("%d", &num);
        for (X = 1; X <= num; X++)
        {
        s= s+X;
        }
        printf ("Sum of 1st %d natural numbers = %d\n", num, s);
    }
```

OUTPUT:

```
Enter the integer
65
Sum of 1st 65 natural numbers = 2145
```

CONCLUSION:

We got the expected result. We can also use the mathematical formula n*(n+1)/2

23)

TITLE:

FACTORIAL

OBJECTIVE:

Write a c program to find factorial of number using recursion.

EXPLANATION:

n>=0, factorial of number n!=1*2*3*4*5*6*7*8*......n

PSEUDO CODE:

```
Recursion (n)
{
If n==0 or n ==1
Return 1
Return n* recursion (n-1)
END
```

CODE:

```
  #include<stdio.h>
long int Number(int n);
int main() {
        int n;
        printf("Enter a natural number: ");
        scanf("%d",&n);
        printf("Factorial of %d = %ld", n, Number(n));
        return 0;
}
long int Number(int n) {
        if (n>=1)
        return n*Number(n-1);
        else
        return 1;
}
```

OUTPUT:

```
Enter a natural number: 56
Factorial of 56 = 6908521828386340864
```

CONCLUSION:

We got the expected result. The time complexity is taken O(n*).

24)

TITLE:

FIBONACCI

OBJECTIVE:

Write a c program to generate the fibonacci series.

EXPLANATION:

For finding the fibonacci series Fn=Fn-1+ Fn-2.

Pseudo code:

```
x=0, y=0, z=0
For i between 0 to n:
{
  z=x+y, x=y, y=z
  Print z
}
END
```

Code:

```
#include <stdio.h>
void fibSeries(int n){
 int i, x= 0, y=1,z;
 for(i=0;i<n;i++){
  z = x+y;
  x = y;
  y = z;
  printf("%d\n",z);
 }
}
int main(int argc, char const *argv[])
{
 int n;
 printf("Enter a n\n");
 scanf("%d",&n);
 fibSeries(n);
 return 0;
}
```

OUTPUT:

```
Enter a n
45
1
2
3
5
8
13
21
34
55
89
144
233
377
610
987
1597
2584
4181
6765
10946
17711
28657
46368
75025
121393
196418
317811
514229
832040
1346269
2178309
3524578
```

```
5702887
9227465
14930352
24157817
39088169
63245986
102334155
165580141
267914296
433494437
701408733
1134903170
1836311903
```

CONCLUSION:
We got the expected result. It is O(n).