

**Name:** Mahalakshmi Kethineni

**Roll No:** CH.SC.U4CSE24160

## WEEK-1

### Program-1:

Write a program to write sum of first n natural numbers using user defined function.

### Code:

```
#include <stdio.h>
int sum(int n){
    int sum1,i;
    for(i=0;i<n;i++){
        sum1 = sum1 + i;
    }
    printf("%d",sum1);
}
int main(){
    int n;
    printf("Enter the number of natural numbers:");
    scanf("%d",&n);
    sum(n);
    return 0;
}
```

The space complexity for this code is  $O(1)$ , because the function which I coded uses a constant amount of memory.

## Output:

```
root@amma43:/home/amma/Documents# gcc -o num num.c
root@amma43:/home/amma/Documents# ./num
Enter the n value:4
10
```

## Program-2:

Write a program to find sum of squares of the first natural numbers.

## Code:

```
#include <stdio.h>
int sumSquares(int n){
    int sum1,i;
    for(i=0;i<n;i++){
        sum1 = sum1 + i*i;
    }
    printf("%d",sum1);
}
int main(){
    int n;
    printf("Enter the number of natural numbers:");
    scanf("%d",&n);
    sumSquares(n);
    return 0;
}
```

## Output:

```
root@amma43:/home/amma/Documents# gcc -o squares squares.c
root@amma43:/home/amma/Documents# ./squares

Enter the value of n 5
55
```

The space complexity for this code is  $O(1)$  because the amount of memory does not grow with  $n$ .

## Program-3:

Write a program to find sum of cubes of the first natural numbers.

## Code:

```
#include <stdio.h>
int sumCube(int n){
    int sum1,i;
    for(i=0;i<n;i++){
        sum1 = sum1 + i*i*i;
    }
    printf("%d",sum1);
}
int main(){
    int n;
    printf("Enter the number of natural numbers:");
    scanf("%d",&n);
    sumCube(n);
    return 0;
}
```

## Output:

```
root@amma43:/home/amma/Documents# gcc -o cubes cubes.c
root@amma43:/home/amma/Documents# ./cubes
Enter the value of n:3
36
root@amma43:/home/amma/Documents# |
```

The space complexity for this code is  $O(1)$  because the amount of memory does not grow with  $n$ .

## Program-4:

Write a program to write factorial of an given integer using recursion.

## Code:

```
#include <stdio.h>
int factorial(int a){
    if(a<=0){
        return 1;
    }
    else{
        return a*factorial(a-1);
    }
}
int main(){
    int n;
    printf("Enter the number for factorial:");
    scanf("%d",&n);
    int fact = factorial(n);
    printf("%d",fact);
    return 0;
}
```

## Output:

```
root@amma43:/home/amma/Documents# gcc -o fact fact.c
root@amma43:/home/amma/Documents# ./fact
Enter the number:4
24
```

The space complexity for this code is  $O(n)$  because the recursive call stores  $n$  frames.

## Program-5:

Write a program for transposing a 3x3 matrix.

## Code:

```
#include <stdio.h>
int main(){
    int a[3][3] = {{1,2,3},
                   {4,5,6},
                   {7,8,9}};

    int i,j;
    for(i=0;i<3;i++){
        for(j=0;j<3;j++){
            printf("%d",a[j][i]);
        }
    }
    return 0;
}
```

## Output:

```
root@amma43:/home/amma/Documents# gcc -o matrix matrix.c
root@amma43:/home/amma/Documents# ./matrix
1      4      2
3      9      6
6      6      4
```

The space complexity for this code is  $O(1)$  because the amount of memory does not grow with  $n$ .

## Program-5:

Write a program to find Fibonacci series.

## Code:

```
#include <stdio.h>
int main() {
    int n, a = 0, b = 1, c;
    printf("Enter number of numbers: ");
    scanf("%d", &n);
    printf("Fibonacci Series: ");
    for (int i = 0; i < n; i++) {
        printf("%d ", a);
        c = a + b;
        a = b;
        b = c;
    }
    return 0;
}
```

## Output:

```
root@amma43:/home/amma/Documents# gcc -o fibonacci fibonacci.c
root@amma43:/home/amma/Documents# ./fibonacci
0      1      1      2      3      5      8      13      21      34      55
```

The space complexity for this code is  $O(1)$  because the amount of memory does not grow with  $n$ .