



SCHOOL OF
COMPUTING

Nithiesh kumarM
CH.SC.U4CSE24230

Week – 1

Date - 29/11/2025

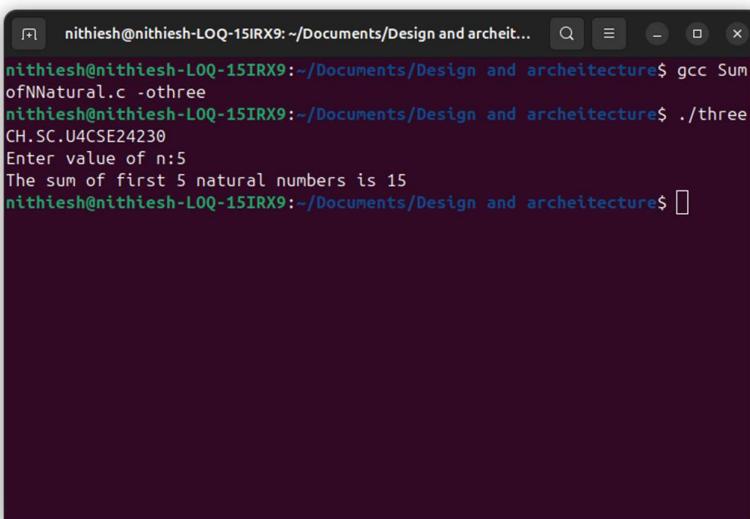
Design and Analysis of Algorithm(23CSE211)

1. Write a program to find sum of first n natural numbers using user defined functions

Code:

```
#include<stdio.h>
int sum(int n){
int sum=0;
for(int i=1;i<n+1;i++){
sum+=i;
}
return sum;
}
int main(){
printf("CH.SC.U4CSE24230\n");
int n;
printf("Enter value of n:");
scanf("%d",&n);
printf("The sum of first %d natural numbers is %d\n",n,sum(n));
}
```

Output:



A screenshot of a terminal window titled "nithiesh@nithiesh-LOQ-15IRX9: ~/Documents/Design and archeit...". The terminal shows the following command-line interaction:

```
nithiesh@nithiesh-LOQ-15IRX9:~/Documents/Design and archeit...$ gcc Sum
ofNNatural.c -othree
nithiesh@nithiesh-LOQ-15IRX9:~/Documents/Design and archeit...$ ./three
CH.SC.U4CSE24230
Enter value of n:5
The sum of first 5 natural numbers is 15
nithiesh@nithiesh-LOQ-15IRX9:~/Documents/Design and archeit...$
```

Space Complexity:

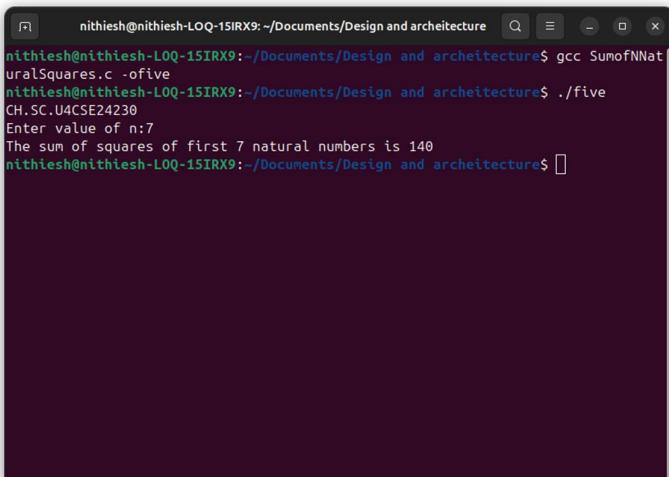
Space Complexity O(1) 3 variables

Justification:

In main(): only 1 int variable n
In sum():int variables - sum,return
so the worst case is O(1)
space used 12 bytes

2. Write a program to find sum of squares of first n natural numbers**Code:**

```
#include<stdio.h>
int main(){
int n;
int sum=0;
printf("CH.SC.U4CSE24230\n");
printf("Enter value of n:");
scanf("%d",&n);
for(int i=1;i<n+1;i++){
sum+=i*i;
}
printf("The sum of squares of first %d natural numbers is
%d\n",n,sum);
}
```

Output:

```
nithiesh@nithiesh-LOQ-15IRX9:~/Documents/Design and archeitecture$ gcc SumofNNaturalSquares.c -o five
nithiesh@nithiesh-LOQ-15IRX9:~/Documents/Design and archeitecture$ ./five
CH.SC.U4CSE24230
Enter value of n:7
The sum of squares of first 7 natural numbers is 140
nithiesh@nithiesh-LOQ-15IRX9:~/Documents/Design and archeitecture$
```

Space Complexity:

Space Complexity O(1) 2 variables

Justification:

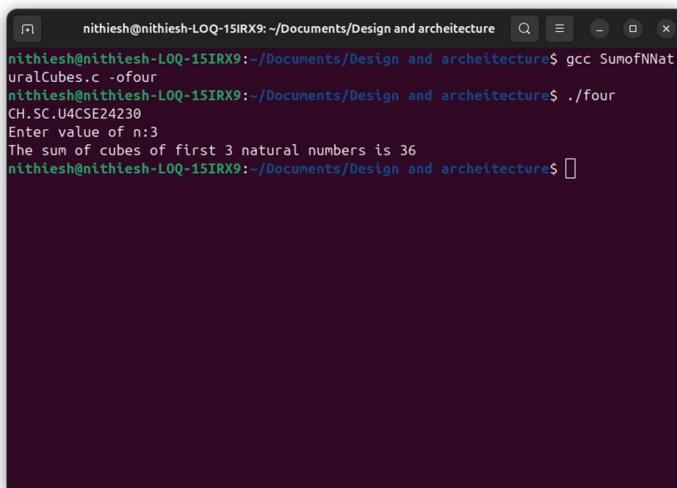
In main(): int variables - n,sum
so the worst case is O(1)
space used 8 bytes

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

Code:

```
#include<stdio.h>
int main(){
printf("CH.SC.U4CSE24230\n");
int n;
int sum=0;
printf("Enter value of n:");
scanf("%d",&n);
for(int i=1;i<n+1;i++){
sum+=i*i*i;
}
printf("The sum of cubes of first %d natural numbers is
%d\n",n,sum);
}
```

Output:



A terminal window titled "nithiesh@nithiesh-LOQ-15IRX9: ~/Documents/Design and archeitecture". The window shows the following command-line interaction:

```
nithiesh@nithiesh-LOQ-15IRX9:~/Documents/Design and archeitecture$ gcc SumofNNaturalCubes.c -ofour
nithiesh@nithiesh-LOQ-15IRX9:~/Documents/Design and archeitecture$ ./four
CH.SC.U4CSE24230
Enter value of n:3
The sum of cubes of first 3 natural numbers is 36
nithiesh@nithiesh-LOQ-15IRX9:~/Documents/Design and archeitecture$
```

Space Complexity:

Space Complexity O(1) 2 variables

Justification:

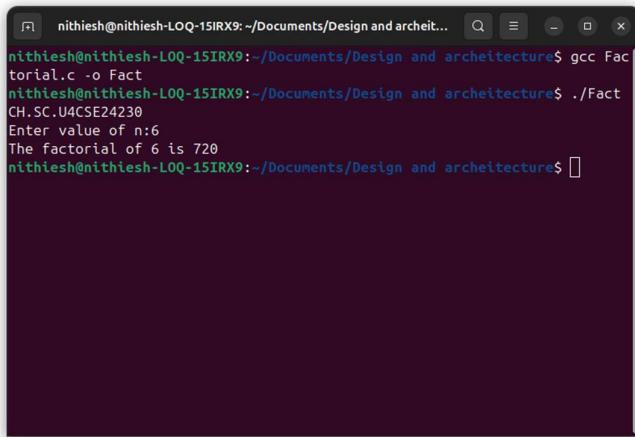
In main(): int variables - n,sum
so the worst case is O(1)
space used 8 bytes

4. Write a program to find factorial of the given integer using recursion

Code:

```
#include<stdio.h>
int factorial(int n){
if(n==1){
return 1;
}
else{
return n*factorial(n-1);
}
}
int main(){
int n;
printf("CH.SC.U4CSE24230\n");
printf("Enter value of n:");
scanf("%d",&n);
printf("The factorial of %d is %d\n",n,factorial(n));
}
```

Output:



A terminal window showing the execution of a C program. The user has typed 'gcc Factorial.c -o Fact' to compile the file 'Factorial.c' into an executable 'Fact'. They then run the command './Fact CH.SC.U4CSE24230'. The program prompts for an input value 'n', which is entered as '6'. The output shows the factorial of 6 is 720.

```
nithiesh@nithiesh-LOQ-15IRX9:~/Documents/Design and archeit...$ gcc Factorial.c -o Fact
nithiesh@nithiesh-LOQ-15IRX9:~/Documents/Design and archeit...$ ./Fact
CH.SC.U4CSE24230
Enter value of n:6
The factorial of 6 is 720
nithiesh@nithiesh-LOQ-15IRX9:~/Documents/Design and archeit...$
```

Space Complexity:

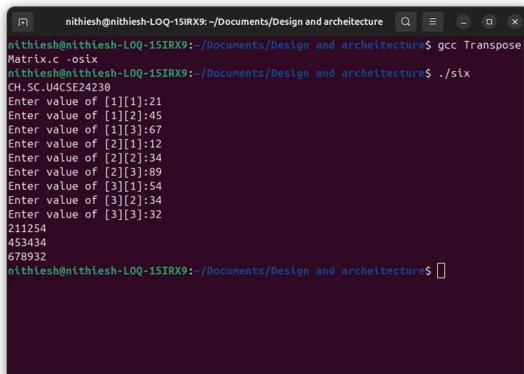
Space Complexity O(n) 1 variable

Justification:

In main(): only 1 int variable n
In factorial():int variable - return
so the worst case is O(n) as the value is returned n times.
space used 4+4n bytes

5. Write a program to for transposing a 3x3 matrix**Code:**

```
#include<stdio.h>
int main(){
printf("CH.SC.U4CSE24230\n");
int mat[3][3]={{0,0,0},{0,0,0},{0,0,0}};
int trans[3][3]={{0,0,0},{0,0,0},{0,0,0}};
for(int i=0;i<3;i++){
for(int j=0;j<3;j++){
printf("Enter value of [%d][%d]:",i+1,j+1);
scanf("%d",&mat[i][j]);
}
}
for(int i=0;i<3;i++){
for(int j=0;j<3;j++){
trans[j][i]=mat[i][j];
}
}
for(int i=0;i<3;i++){
for(int j=0;j<3;j++){
printf("%d",trans[i][j]);
}
printf("\n");
}
}
```

Output:

A terminal window showing the execution of a C program to transpose a 3x3 matrix. The program prompts the user to enter values for a 3x3 matrix, stores them in 'mat', and then prints the transpose stored in 'trans'. The terminal shows the command 'gcc TransposeMatrix.c -o six' and the output of the program.

```
nithiesh@nithiesh-LQ0-15IRX9:~/Documents/Design and architecture$ gcc TransposeMatrix.c -o six
nithiesh@nithiesh-LQ0-15IRX9:~/Documents/Design and architecture$ ./six
CH.SC.U4CSE24230
Enter value of [1][1]:21
Enter value of [1][2]:45
Enter value of [1][3]:67
Enter value of [2][1]:12
Enter value of [2][2]:34
Enter value of [2][3]:89
Enter value of [3][1]:54
Enter value of [3][2]:34
Enter value of [3][3]:32
211254
453434
678932
nithiesh@nithiesh-LQ0-15IRX9:~/Documents/Design and architecture$
```

Space Complexity:

Space Complexity O(1) 2 arrays

Justification:

In main(): Arrays - mat[3][3], trans[3][3].

The total space is fixed (constant) regardless of any input, as the matrix size is hardcoded to 3x3.

The worst case is O(1).

space used 72 bytes

6. Write a program to find Fibonacci series

Code:

```
#include <stdio.h>
int main() {
    int n;
    printf("CH.SC.U4CSE24230\n");
    printf("Enter the number of Fibonacci terms to generate: ");
    if (scanf("%d", &n) != 1) {
        printf("Invalid input. Please enter an integer.\n");
        return 1;
    }
    if (n <= 0) {
        printf("Please enter a positive integer greater than 0.\n");
        return 0;
    }
    int t1 = 0;
    int t2 = 1;
    int nextTerm;
    printf("Fibonacci Series (first %d terms):\n", n);
    if (n >= 1) {
        printf("%d", t1);
    }
    if (n >= 2) {
        printf(", %d", t2);
    }
    for (int i = 3; i <= n; ++i) {
        nextTerm = t1 + t2;
        if (nextTerm < t2) {
            printf("\n\n(Note: Integer overflow occurred at term %d. Output may
be inaccurate from this point.)\n", i);
            break;
        }
        printf(", %d", nextTerm);
        t1 = t2;
        t2 = nextTerm;
    }
}
```

Output:

```
nithiesh@nithiesh-LOQ-15IRX9:~/Documents/Design and archeit...      
nithiesh@nithiesh-LOQ-15IRX9:~/Documents/Design and archeitecture$ gcc Fib  
onacci.c  
nithiesh@nithiesh-LOQ-15IRX9:~/Documents/Design and archeitecture$ gcc Fib  
onacci.c -o Fib  
nithiesh@nithiesh-LOQ-15IRX9:~/Documents/Design and archeitecture$ ./Fib  
CH.SC.U4CSE24230  
Enter the number of Fibonacci terms to generate: 4  
Fibonacci Series (first 4 terms):  
0, 1, 1, 2  
nithiesh@nithiesh-LOQ-15IRX9:~/Documents/Design and archeitecture$ 
```

Space Complexity:

Space Complexity O(1) 4 variables

Justification:

In main(): int variables - n, t1, t2, nextTerm.

The total space is fixed (constant) regardless of the input value 'n'.

The worst case is O(1).

space used 16 bytes