

Kishore S

CH.SC.U4CSE24222

Week – 1

Date - 29/11/2025

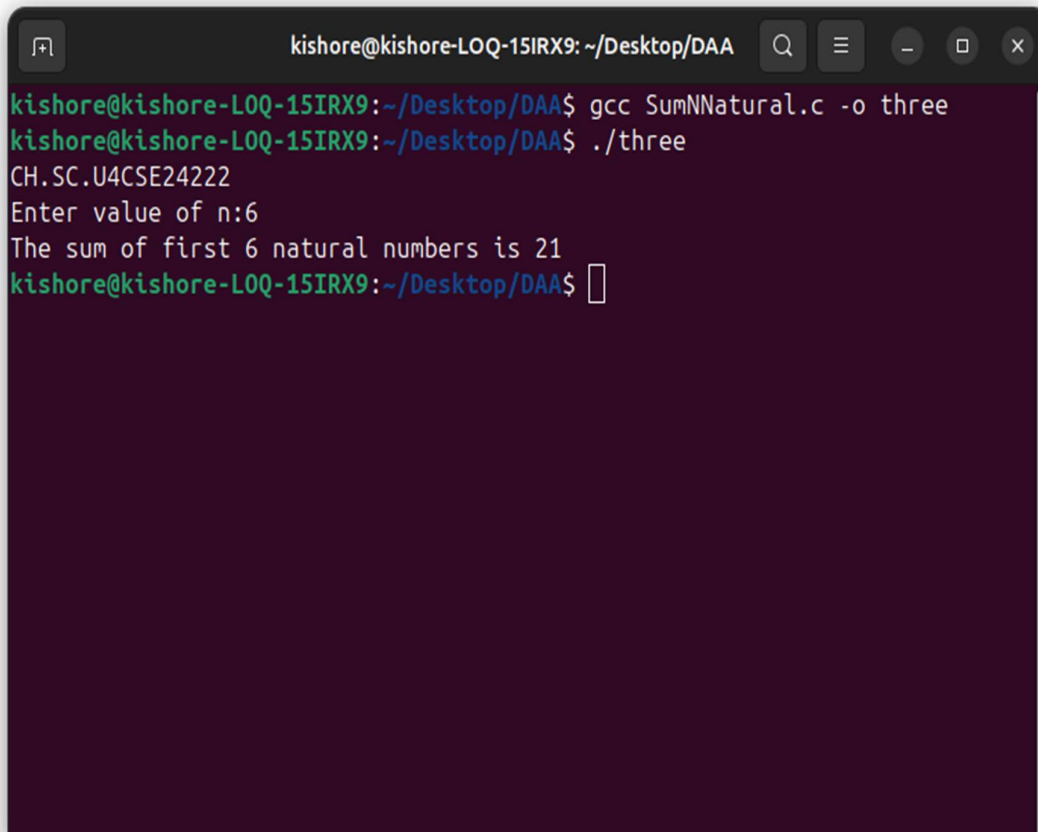
Design and Analysis of Algorithm(23CSE211)

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.U4CSE24222\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 terminal window with a dark purple background and white text. The window title is "kishore@kishore-LOQ-15IRX9: ~/Desktop/DAA". The terminal shows the following commands and output:

```
kishore@kishore-LOQ-15IRX9:~/Desktop/DAA$ gcc SumNNatural.c -o three
kishore@kishore-LOQ-15IRX9:~/Desktop/DAA$ ./three
CH.SC.U4CSE24222
Enter value of n:6
The sum of first 6 natural numbers is 21
kishore@kishore-LOQ-15IRX9:~/Desktop/DAA$
```

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 $O(1)$

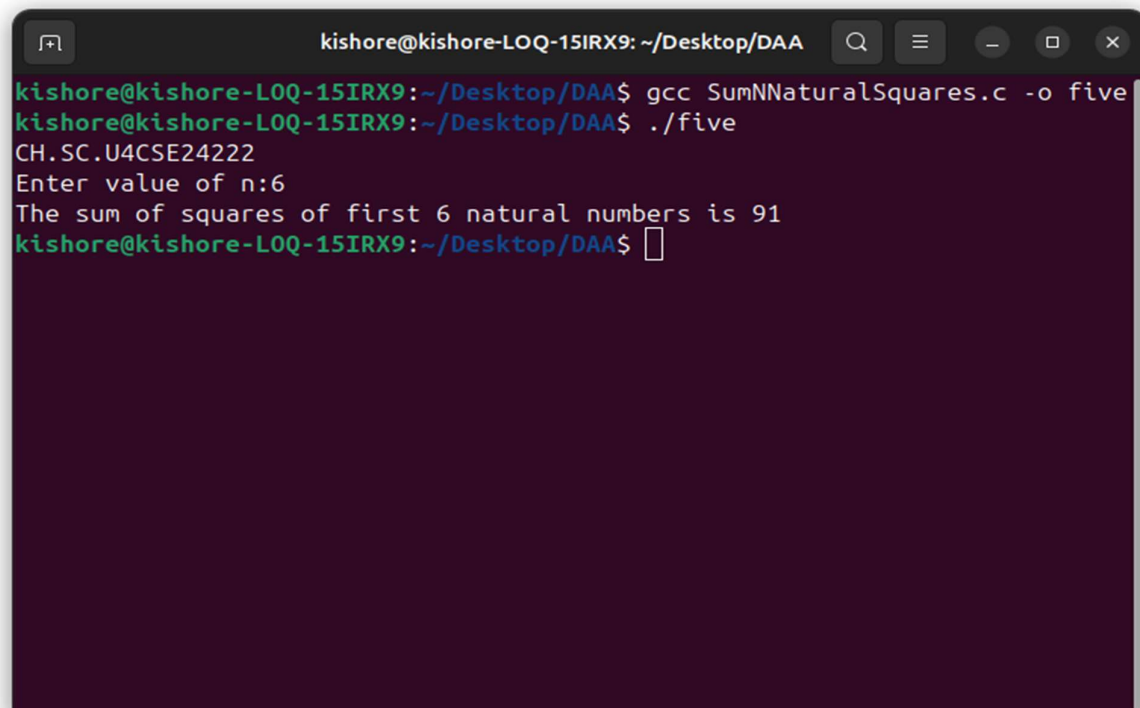
space used 12 bytes

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.U4CSE24222\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:

A terminal window with a dark purple background and white text. The window title is "kishore@kishore-LOQ-15IRX9: ~/Desktop/DAA". The terminal shows the following commands and output:
kishore@kishore-LOQ-15IRX9:~/Desktop/DAA\$ gcc SumNNaturalSquares.c -o five
kishore@kishore-LOQ-15IRX9:~/Desktop/DAA\$./five
CH.SC.U4CSE24222
Enter value of n:6
The sum of squares of first 6 natural numbers is 91
kishore@kishore-LOQ-15IRX9:~/Desktop/DAA\$
The terminal window has standard Linux window controls (minimize, maximize, close) in the top right corner.

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.U4CSE24222\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 with a dark purple background. The title bar shows the user 'kishore' on a machine named 'kishore-LOQ-15IRX9' in the directory '~/Desktop/DAA'. The terminal shows the following commands and output:
1. Command: gcc SumNNaturalCubes.c -o four
2. Command: ./four
3. Output: CH.SC.U4CSE24222
4. Prompt: Enter value of n:
5. Input: 5
6. Output: The sum of cubes of first 5 natural numbers is 225
7. Prompt: kishore@kishore-LOQ-15IRX9:~/Desktop/DAA\$
The cursor is visible at the end of the last prompt line.

```
kishore@kishore-LOQ-15IRX9: ~/Desktop/DAA
kishore@kishore-LOQ-15IRX9:~/Desktop/DAA$ gcc SumNNaturalCubes.c -o four
kishore@kishore-LOQ-15IRX9:~/Desktop/DAA$ ./four
CH.SC.U4CSE24222
Enter value of n:5
The sum of cubes of first 5 natural numbers is 225
kishore@kishore-LOQ-15IRX9:~/Desktop/DAA$
```

Space Complexity:

Space Complexity $O(1)$ 2 variables

Justification:

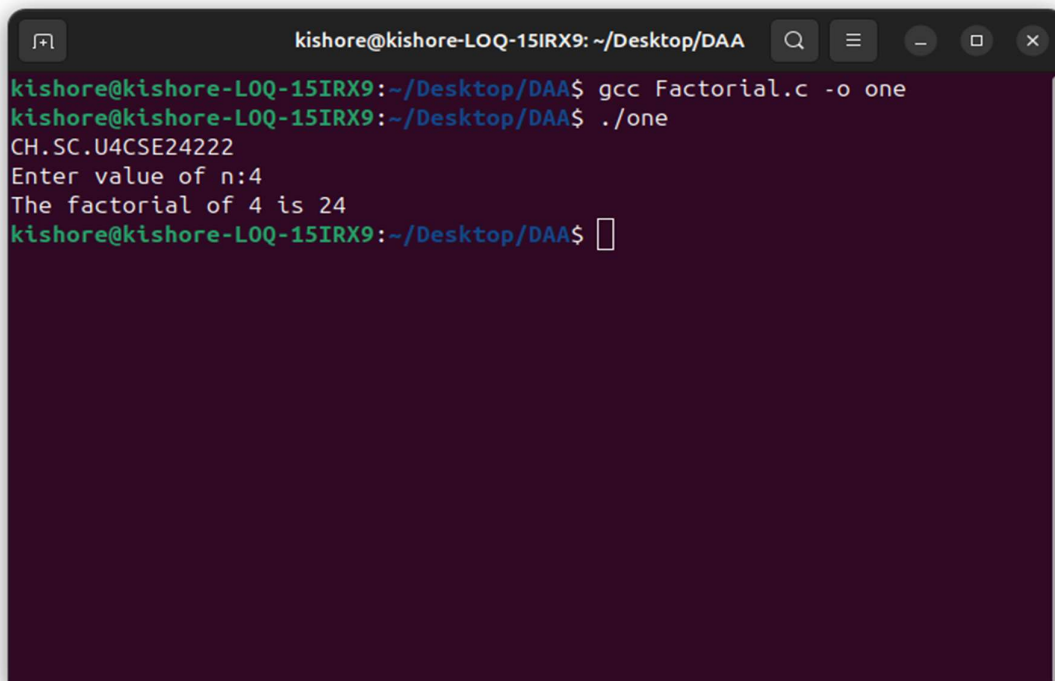
In main(): int variables - n, sum

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.U4CSE24113\n");
printf("Enter value of n:");
scanf("%d",&n);
printf("The factorial of %d is %d\n",n,factorial(n));
}
```

Output:A terminal window with a dark purple background. The title bar shows the user 'kishore' on machine 'kishore-LOQ-15IRX9' in the directory '~/Desktop/DAA'. The terminal shows the following commands and output:
kishore@kishore-LOQ-15IRX9:~/Desktop/DAA\$ gcc Factorial.c -o one
kishore@kishore-LOQ-15IRX9:~/Desktop/DAA\$./one
CH.SC.U4CSE24222
Enter value of n:4
The factorial of 4 is 24
kishore@kishore-LOQ-15IRX9:~/Desktop/DAA\$
The prompt character is a white square.

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 transpose a 3x3 matrix**Code:**

```
#include<stdio.h>
int main(){
printf("CH.SC.U4CSE24113\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:

```
kishore@kishore-LOQ-15IRX9: ~/Desktop/DAA
kishore@kishore-LOQ-15IRX9:~/Desktop/DAA$ gcc TransposeMatrix.c -o six
kishore@kishore-LOQ-15IRX9:~/Desktop/DAA$ ./six
CH.SC.U4CSE24222
Enter value of [1][1]:4
Enter value of [1][2]:4
Enter value of [1][3]:5
Enter value of [2][1]:6
Enter value of [2][2]:2
Enter value of [2][3]:9
Enter value of [3][1]:7
Enter value of [3][2]:9
Enter value of [3][3]:1
467
429
591
kishore@kishore-LOQ-15IRX9:~/Desktop/DAA$
```

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 $O(1)$.

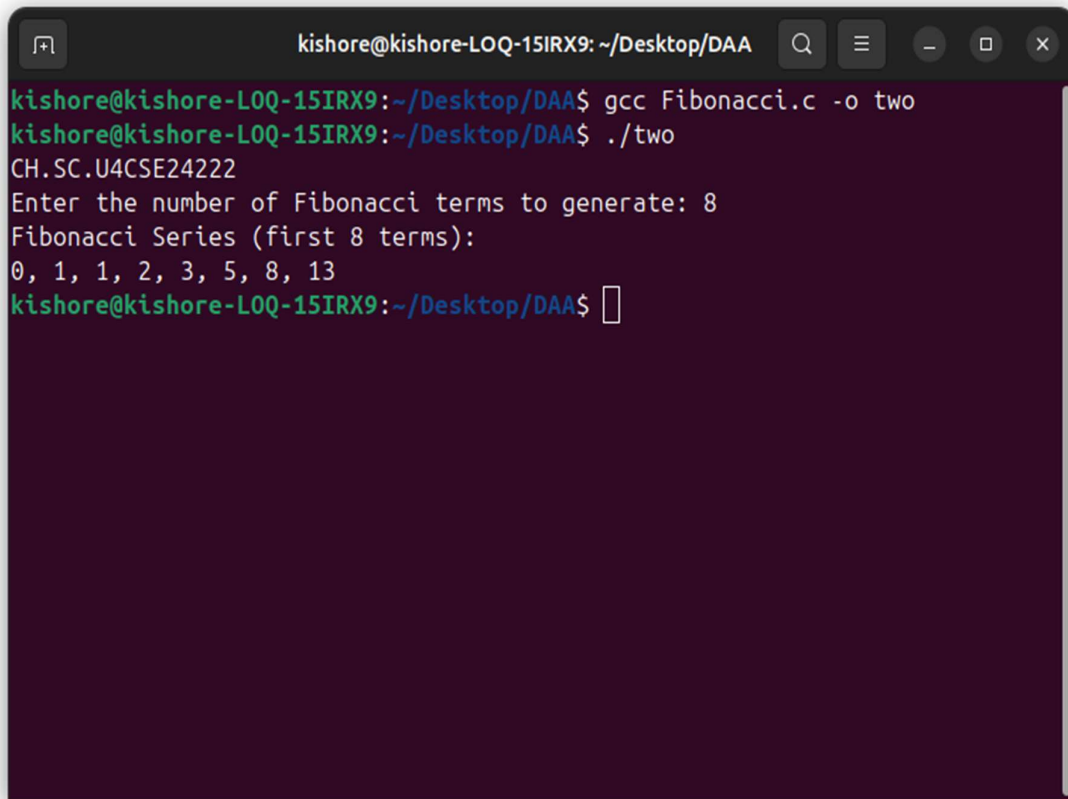
space used 72 bytes

6. Write a program to find Fibonacci series

Code:

```
#include <stdio.h>
int main() {
    int n;
    printf("CH.SC.U4CSE24113\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:



```
kishore@kishore-LOQ-15IRX9: ~/Desktop/DAA
kishore@kishore-LOQ-15IRX9:~/Desktop/DAA$ gcc Fibonacci.c -o two
kishore@kishore-LOQ-15IRX9:~/Desktop/DAA$ ./two
CH.SC.U4CSE24222
Enter the number of Fibonacci terms to generate: 8
Fibonacci Series (first 8 terms):
0, 1, 1, 2, 3, 5, 8, 13
kishore@kishore-LOQ-15IRX9:~/Desktop/DAA$
```

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