AMRITA VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY UNDER SECTION 3 OF UGC ACT, 1956
SCHOOL OF COMPUTING

Devadharshan.S

CH.SC.U4CSE24113

Week – 1

Date - 27/11/2025

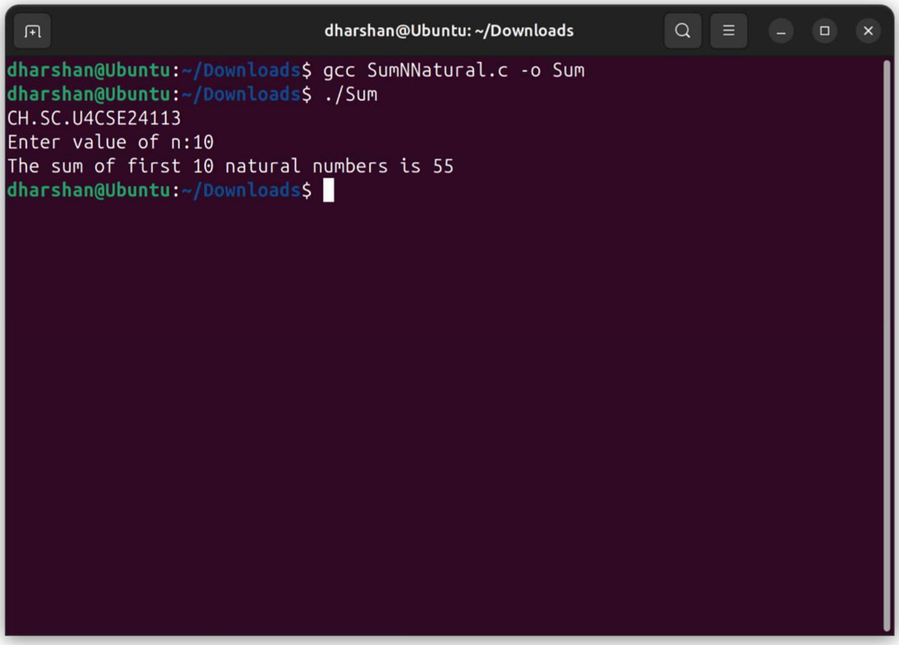
Design and Analysis of Algorithm(23CSE211)

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.U4CSE24113\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:



```
dharshan@Ubuntu:~/Downloads$ gcc SumNNatural.c -o Sum
dharshan@Ubuntu:~/Downloads$ ./Sum
CH.SC.U4CSE24113
Enter value of n:10
The sum of first 10 natural numbers is 55
dharshan@Ubuntu:~/Downloads$ 
```

Space Complexity:

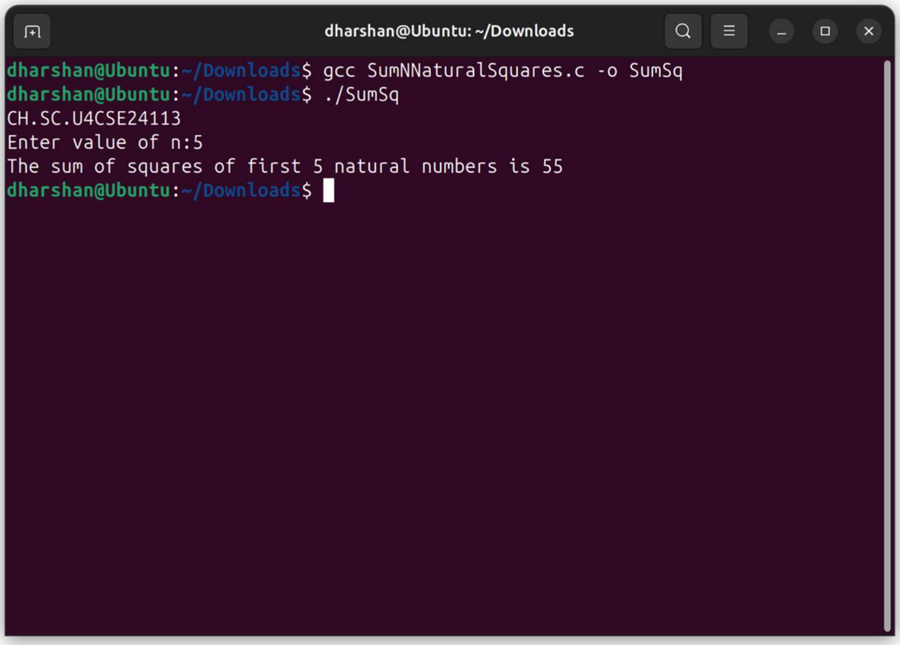
Space Complexity O(1) 3 variables
Justification:
In main(): only 1 int variable n
In sum():int variables - sum,return
so the worst case is O(1)
space used 12 bytes

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.U4CSE24113\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:



```
dharshan@Ubuntu:~/Downloads$ gcc SumNNaturalSquares.c -o SumSq
dharshan@Ubuntu:~/Downloads$ ./SumSq
CH.SC.U4CSE24113
Enter value of n:5
The sum of squares of first 5 natural numbers is 55
dharshan@Ubuntu:~/Downloads$ 
```

Space Complexity:

Space Complexity O(1) 2 variables

Justification:

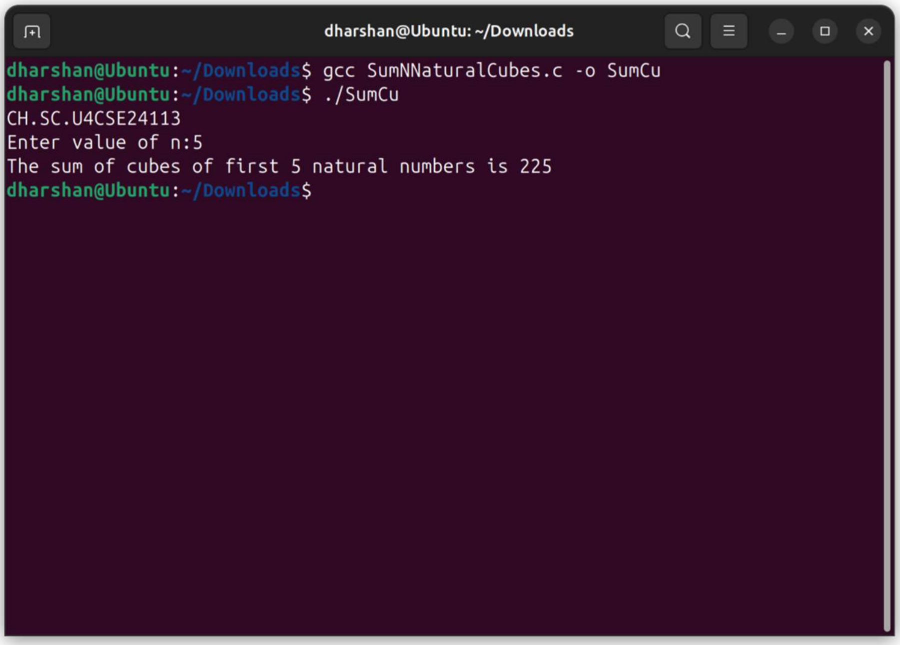
In main(): int variables - n,sum
so the worst case is O(1)
space used 8 bytes

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

Code:

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



```
dharshan@Ubuntu:~/Downloads$ gcc SumNNaturalCubes.c -o SumCu
dharshan@Ubuntu:~/Downloads$ ./SumCu
CH.SC.U4CSE24113
Enter value of n:5
The sum of cubes of first 5 natural numbers is 225
dharshan@Ubuntu:~/Downloads$
```

Space Complexity:

Space Complexity O(1) 2 variables

Justification:

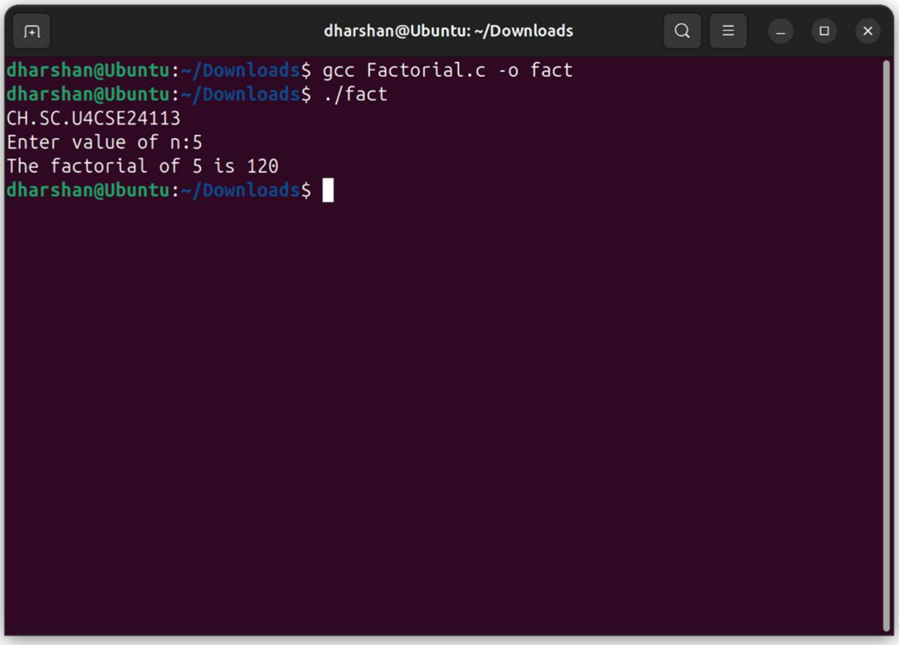
In main(): int variables - n,sum
so the worst case is O(1)
space used 8 bytes

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:



```
dharshan@Ubuntu:~/Downloads$ gcc Factorial.c -o fact
dharshan@Ubuntu:~/Downloads$ ./fact
CH.SC.U4CSE24113
Enter value of n:5
The factorial of 5 is 120
dharshan@Ubuntu:~/Downloads$
```

Space Complexity:

Space Complexity O(n) 1 variable

Justification:

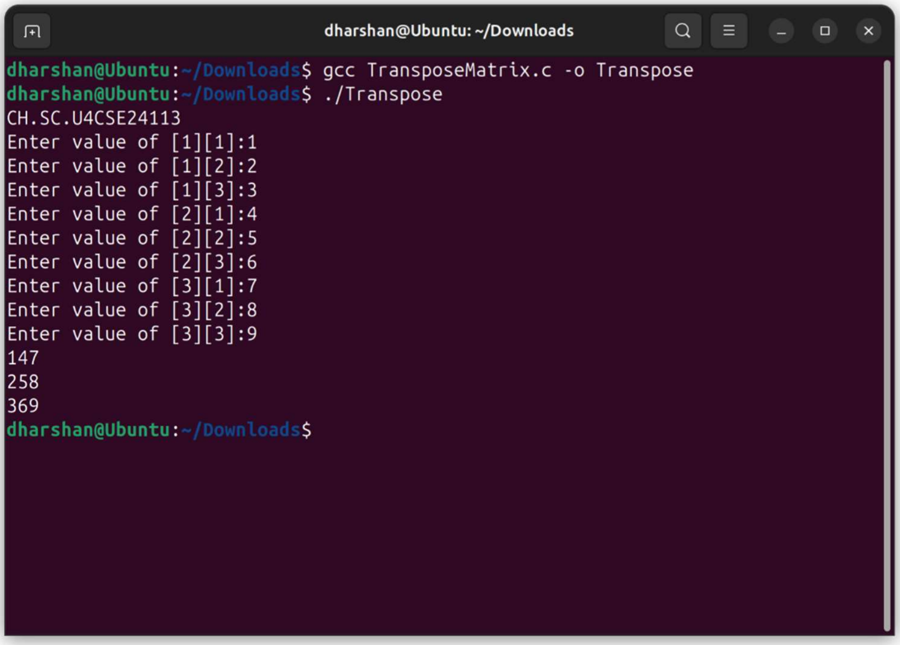
In main(): only 1 int variable n
In factorial():int variable - return
so the worst case is O(n) as the value is returned n times.
space used 4+4n bytes

5.Write a program to for transposing a 3x3 matrix

Code:

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



```
dharshan@Ubuntu:~/Downloads$ gcc TransposeMatrix.c -o Transpose
dharshan@Ubuntu:~/Downloads$ ./Transpose
CH.SC.U4CSE24113
Enter value of [1][1]:1
Enter value of [1][2]:2
Enter value of [1][3]:3
Enter value of [2][1]:4
Enter value of [2][2]:5
Enter value of [2][3]:6
Enter value of [3][1]:7
Enter value of [3][2]:8
Enter value of [3][3]:9
147
258
369
dharshan@Ubuntu:~/Downloads$
```

Space Complexity:

Space Complexity O(1) 2 arrays

Justification:

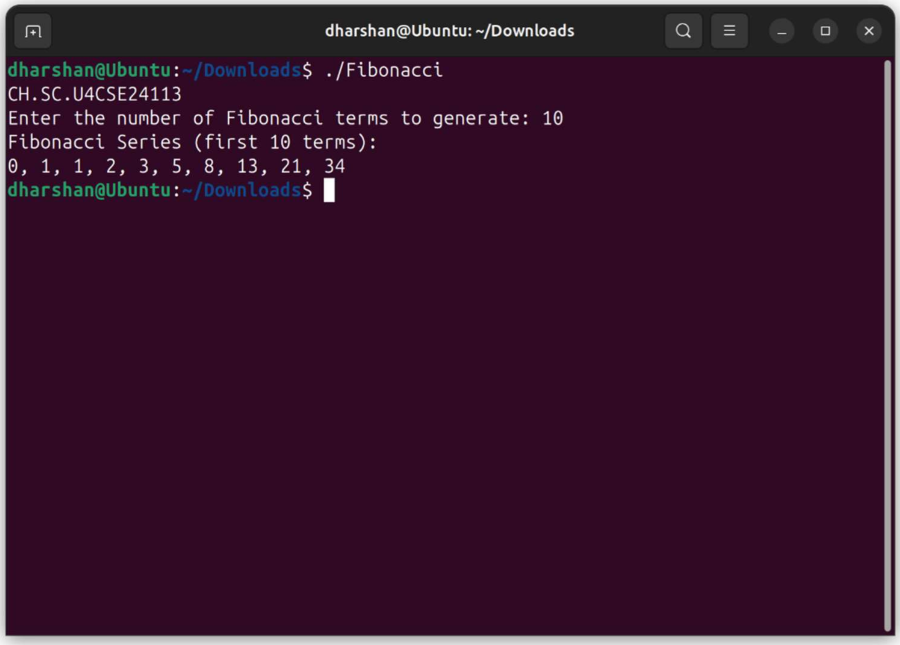
In main(): Arrays - mat[3][3], trans[3][3].
The total space is fixed (constant) regardless of any input, as the matrix size is hardcoded to 3x3.
The worst case is O(1).
space used 72 bytes

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:



```
dharshan@Ubuntu:~/Downloads$ ./Fibonacci
CH.SC.U4CSE24113
Enter the number of Fibonacci terms to generate: 10
Fibonacci Series (first 10 terms):
0, 1, 1, 2, 3, 5, 8, 13, 21, 34
dharshan@Ubuntu:~/Downloads$
```

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