Name: Chetan Kumar G

CSE - B

CH.SC.U4CSE24109

Week -1(27/11/2025)

1.Write a program to find the sum of first N natural number

CODE:

```c
//1.Write a program to find the sum of first N natural number
#include <stdio.h>
void sumOfNaturalNumbers(int n){
int sum=0;
for(int i=1;i<=n;i++){
sum+=i;
}
printf("the sum is= %d\n",sum);
}
int main(){
int n;
printf("Enter the number");
scanf("%d",&n);
sumOfNaturalNumbers(n);

printf("Space Complexity O(1) since the number of variable created is not changing on based on input so it is O(1).\n");
printf("all the variables created does not depend on the input it will be the same for any input so no change on input.\n");
return 0;
}
```

OUTPUT:

```
chetan@amma07:~/DAA/27nov$ gcc p1.c -o p1
chetan@amma07:~/DAA/27nov$ ./p1
Enter the number10
the sum is= 55
Space Complexity O(1) since the number of variable created is not changing on based on input so it is O(1).
all the variables created does not depend on the input it will be the same for any input so no change on input.
```

JUSTIFICATION:

Space Complexity O(1) since the number of variable created is not changing on based on input so it is O(1)

all the variables created does not depend on the input it will be the same for any input so no change on input.

2.Write a program to find the sum of Square first N natural number

CODE:

```c
//2.Write a program to find the sum of Square first N natural number
#include <stdio.h>
void sumOfSqNaturalNumbers(int n){
int sum=0;
for(int i=1;i<=n;i++){
sum=sum+(i*i);
}
printf("the sum is= %d\n",sum);
}
int main(){
int n;
printf("Enter the number");
scanf("%d",&n);
sumOfSqNaturalNumbers(n);

printf("Space Complexity O(1) since the number of variable created is not changing on based on input so it is O(1).\n");
printf("all the variables created does not depend on the input it will be the same for any input so no change on input.\n");
return 0;
}
```

OUTPUT:

```
chetan@amma07:~/DAA/27nov$ gcc p2.c -o p2
chetan@amma07:~/DAA/27nov$ ./p2
Enter the number10
the sum is= 385
Space Complexity O(1) since the number of variable created is not changing on based on input so it is O(1).
all the variables created does not depend on the input it will be the same for any input so no change on input.
```

JUSTIFICATION:

Space Complexity O(1) since the number of variable created is not changing on based on input so it is O(1)

all the variables created does not depend on the input it will be the same for any input so no change on input.

3.Write a program to find the sum of Cube first N natural number

CODE:

```c
//3.Write a program to find the sum of Cube first N natural number
#include <stdio.h>
void sumOfCubeNaturalNumbers(int n){
int sum=0;
for(int i=1;i<=n;i++){
sum=sum+(i*i*i);
}
printf("the sum is= %d\n",sum);
}
int main(){
int n;
printf("Enter the number");
scanf("%d",&n);
sumOfCubeNaturalNumbers(n);

printf("Space Complexity O(1) since the number of variable created is not changing on based on input so it is O(1).\n");
printf("all the variables created does not depend on the input it will be the same for any input so no change on input.\n");
return 0;
}
```

OUTPUT:

```
chetan@amma07:~/DAA/27nov$ gcc p3.c -o p3
chetan@amma07:~/DAA/27nov$ ./p3
Enter the number10
the sum is= 3025
Space Complexity O(1) since the number of variable created is not changing on based on input so it is O(1).
all the variables created does not depend on the input it will be the same for any input so no change on input.
```

JUSTIFICATION:

Space Complexity O(1) since the number of variable created is not changing on based on input so it is O(1)

all the variables created does not depend on the input it will be the same for any input so no change on input.

4.Write a program to find factorial using recursion

CODE:

```c
//4. Write a program to find factorial using recursion
#include <stdio.h>
int factorial(int n){
if(n==1){
return 1;
}
return n*factorial(n-1);
}
int main(){
int n;
printf("Enter the number");
scanf("%d",&n);
int b=factorial(n);
printf("the result=%d\n",b);
printf("Space Complexity O(n) since the number of variable created is depended on the input so the complexity is O(n).");
printf("Since it is recursively calling the function for n times so the variable n will be called n times.");
return 0;
}
```

OUTPUT:

```
chetan@amma07:~/DAA/27nov$ gcc p4.c -o p4
chetan@amma07:~/DAA/27nov$ ./p4
Enter the number10
the result=3628800
Space Complexity O(n) since the number of variable created is depended on the input so the complexity is O(n).Since it is recursively calling the functi
on for n times so the variable n will be called n times.chetan@amma07:~/DAA/27nov$
```

JUSTIFICATION:

Space Complexity O(n) since the number of variable created is depended on the input so the complexity is O(n).

Since it is recursively calling the function for n times so the variable n will be called n times.

5.Write a program to find transpose of a 3*3 Matrix

CODE:

```c
//5.Write a program to find transpose of a 3*3 Matrix
#include <stdio.h>
int main(){
int arr[3][3];
for(int i=0;i<3;i++){
for(int j=0;j<3;j++){
scanf("%d",&arr[i][j]);
}
}

int trans[3][3];
for(int i=0;i<3;i++){
for(int j=0;j<3;j++){
trans[j][i]=arr[i][j];
}
}

for(int i=0;i<3;i++){
for(int j=0;j<3;j++){
printf("%d ",trans[i][j]);
}
printf("\n");
}
printf("Space Complexity O(1) since the number of variable created is not changing on based on input so it is O(1).\n");
printf("all the variables created does not depend on the input it will be the same for any input so no change on input.\n");
return 0;
}
```

OUTPUT:

```
chetan@amma07:~/DAA/27nov$ gcc p5.c -o p5
chetan@amma07:~/DAA/27nov$ ./p5
1 2 3
4 5 6
7 8 9
1 4 7
2 5 8
3 6 9
Space Complexity O(1) since the number of variable created is not changing on based on input so it is O(1).
all the variables created does not depend on the input it will be the same for any input so no change on input.
chetan@amma07:~/DAA/27nov$
```

JUSTIFICATION:

Space Complexity O(1) since the number of variable created is not changing on based on input so it is O(1)

all the variables created does not depend on the input it will be the same for any input so no change on input.

Suppose the matrix's dimensions were not fixed and where arr[m][n] then the complexity would have been O(n*m).

6.Write a program to find Fibonacci series

CODE:

```c
//6. Write a program to find Fibonacci series
#include <stdio.h>
int main(){
int a=0, b=1, sum,n;
printf("Enter a number");
scanf("%d",&n);
printf("%d %d ",a,b);
for(int i=1;i<n-1;i++){
sum=a+b;
a=b;
b=sum;
printf("%d ",sum);
}

printf("Space Complexity O(1) since the number of variable created is not changing on based on input so it is O(1).\n");
printf("all the variables created does not depend on the input it will be the same for any input so no change on input.\n");
return 0;

}
```

OUTPUT:

```
chetan@amma07:~/DAA/27nov$ gcc p6.c -o p6
chetan@amma07:~/DAA/27nov$ ./p6
Enter a number10
0 1 1 2 3 5 8 13 21 34 Space Complexity O(1) since the number of variable created is not changing on based on input so it is O(1).
all the variables created does not depend on the input it will be the same for any input so no change on input.
chetan@amma07:~/DAA/27nov$
```

JUSTIFICATION:

Space Complexity O(1) since the number of variable created is not changing on based on input so it is O(1)

all the variables created does not depend on the input it will be the same for any input so no change on input.