

**SCHOOL OF
COMPUTING**

DESIGN AND ANALYSIS OF ALGORITHMS

LAB WORKBOOK

WEEK - 1

NAME : B.S.Chenthil Hari

ROLL NUMBER : CH.SC.U4CSE24103

CLASS : CSE-B

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

CODE:

```
[*] first.c ×
1  #include<stdio.h>
2  int sumofn(int n){
3      int i;
4      int sum=0;
5      for(i=0;i<=n;i++){
6          sum+=i;
7      }
8      return sum;
9  }
10 int main(){
11     int n,result;
12     printf("Enter the number: ");
13     scanf("%d",&n);
14     printf("The sum of the %d natural numbers is: ",n);
15     result=sumofn(n);
16     printf("%d\n",result);
17     return 0;
18 }
19
```

OUTPUT:

```
C:\Users\chent\Downloads\fir × + v
Enter the number: 5
The sum of the 5 natural numbers is: 15

-----
Process exited after 1.569 seconds with return value 0
Press any key to continue . . .
```

Space Complexity: $O(1)$

Justification:

In main()

int n → 4 bytes

int sum1 → 4 bytes

Total in main() = 8 bytes

In sum()

int sum → 4 bytes

int i → 4 bytes

Total in sum() = 8 bytes

Now the total bytes required for this program is 16 bytes, which is constant.

Question 2: Write a program to find sum of squares of the first n natural numbers.

CODE:

```
1  #include<stdio.h>
2  int sumofn(int n){
3      int i;
4      int sum=0;
5      for(i=0;i<=n;i++){
6          sum+=i*i;
7      }
8      return sum;
9  }
10 int main(){
11     int n,result;
12     printf("Enter the number: ");
13     scanf("%d",&n);
14     printf("The sum of square of the %d natural numbers is: ",n);
15     result=sumofn(n);
16     printf("%d\n",result);
17     return 0;
18 }
```

OUTPUT:

```
C:\Users\chent\Downloads\se  X + v
Enter the number: 6
The sum of square of the 6 natural numbers is: 91

-----
Process exited after 1.498 seconds with return value 0
Press any key to continue . . .
```

Space Complexity: $O(1)$

Justification:

In main()

int n \rightarrow 4 bytes

int sum \rightarrow 4 bytes

int i (loop variable) \rightarrow 4 bytes

Total space in main = 12 bytes

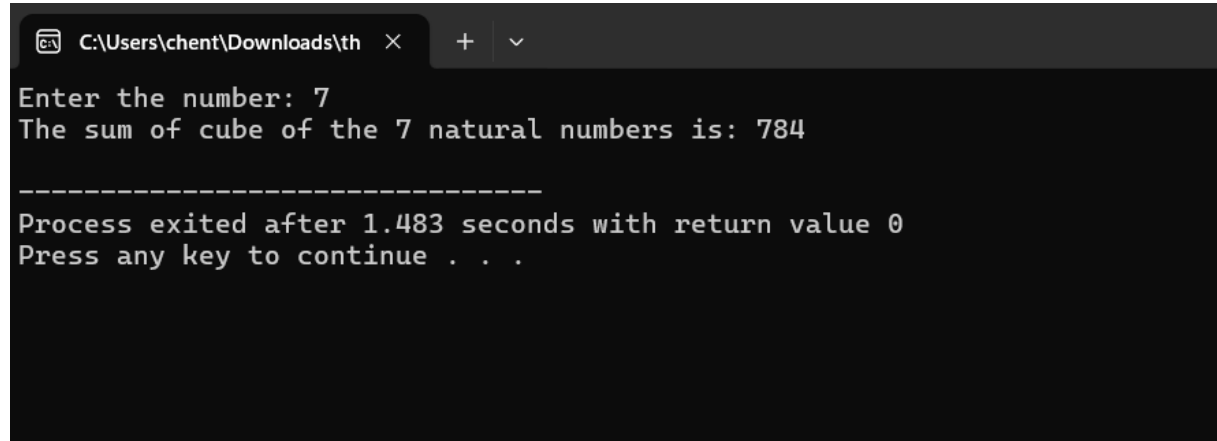
Now, the total space required for the program to execute is always 12 bytes, which is a constant.

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

CODE:

```
third.c  X
1  #include<stdio.h>
2  int sumofn(int n){
3      int i;
4      int sum=0;
5      for(i=0;i<=n;i++){
6          sum+=i*i*i;
7      }
8      return sum;
9  }
10 int main(){
11     int n,result;
12     printf("Enter the number: ");
13     scanf("%d",&n);
14     printf("The sum of cube of the %d natural numbers is: ",n);
15     result=sumofn(n);
16     printf("%d\n",result);
17     return 0;
18 }
```

OUTPUT:

A screenshot of a web browser window. The address bar shows the path 'C:\Users\chent\Downloads\th'. The main content area displays the output of a program. It starts with the prompt 'Enter the number: 7', followed by the result 'The sum of cube of the 7 natural numbers is: 784'. A horizontal line of dashes separates this from the final output: 'Process exited after 1.483 seconds with return value 0' and 'Press any key to continue . . .'.

```
Enter the number: 7
The sum of cube of the 7 natural numbers is: 784

-----
Process exited after 1.483 seconds with return value 0
Press any key to continue . . .
```

Space Complexity: $O(1)$

Justification:

In main()

int n \rightarrow 4 bytes

int sum \rightarrow 4 bytes

int i (loop variable) \rightarrow 4 bytes

Total space used in main() = 12 bytes

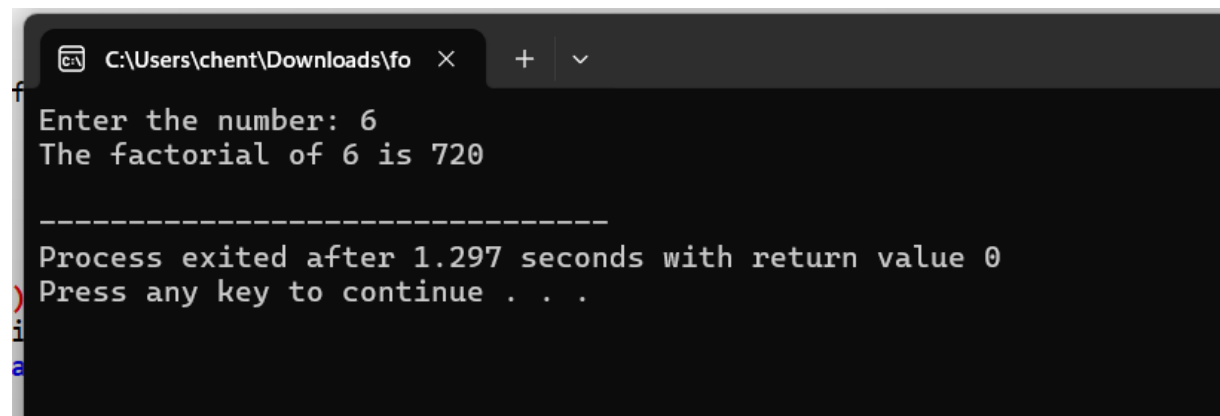
Now, the total space required for the program to execute is always 12 bytes, which is a constant.

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

CODE:

```
1  #include<stdio.h>
2  int factorial(int n){
3      if(n==1){
4          return 1;
5      }
6      else{
7          return n*factorial(n-1);
8      }
9  }
10 int main(){
11     int n,result;
12     printf("Enter the number: ");
13     scanf("%d",&n);
14     result=factorial(n);
15     printf("The factorial of %d is %d\n",n,result);
16     return 0;
17 }
```

OUTPUT:



```
C:\Users\chent\Downloads\fo x + v
f
Enter the number: 6
The factorial of 6 is 720
-----
Process exited after 1.297 seconds with return value 0
Press any key to continue . . .
)
i
a
```

Space Complexity: $O(n)$

Justification:

In main()

int n \rightarrow 4 bytes

int fact \rightarrow 4 bytes

Space in main = **8 bytes** (constant)

In factorial()

Each call to factorial(n) creates:

int n \rightarrow 4 bytes

This function is recursive, so for input **n**, the function calls **n** times.

Space required for the factorial function is $4 * n$ bytes.

The equation for the Space Complexity is $4*n + 8$. The order of Space Complexity is **n**.

Question 5: Write a program for transposing a 3 x 3 matrix.

CODE:

```
1  #include<stdio.h>
2  int main(){
3      int n,result,i,j;
4      int matrix[3][3];
5      for(i=0;i<3;i++){
6          for(j=0;j<3;j++){
7              printf("Enter row %d column %d element: ",i+1,j+1);
8              scanf("%d",&matrix[i][j]);
9          }
10     }
11     printf("\n");
12     printf("You've Entered Matrix \n");
13     for(i=0;i<3;i++){
14         for(j=0;j<3;j++){
15             printf("%d ",matrix[i][j]);
16         }
17         printf("\n");
18     }
19     printf("\n\n");
20     printf("Transpose of The Above Matrix is: \n");
21     int transpose[3][3];
22     for(i=0;i<3;i++){
23         for(j=0;j<3;j++){
24             transpose[j][i]=matrix[i][j];
25         }
26     }
27     for(i=0;i<3;i++){
28         for(j=0;j<3;j++){
29             printf("%d ",transpose[i][j]);
30         }
31         printf("\n");
32     }
33     return 0;
34 }
```

OUTPUT:


```
C:\Users\chent\Downloads\fff x + v
Enter row 1 column 1 element: 4
Enter row 1 column 2 element: 2
Enter row 1 column 3 element: 5
Enter row 2 column 1 element: 6
Enter row 2 column 2 element: 7
Enter row 2 column 3 element: 8
Enter row 3 column 1 element: 9
Enter row 3 column 2 element: 7
Enter row 3 column 3 element: 5

You've Entered Matrix
4 2 5
6 7 8
9 7 5

Transpose of The Above Matrix is:
4 6 9
2 7 7
5 8 5

-----
Process exited after 4.158 seconds with return value 0
Press any key to continue . . .
```

Space Complexity: $O(1)$

Justification:

In main()I

int R \rightarrow 4 bytes

int C \rightarrow 4 bytes

Loop variables i and j $\rightarrow 4 + 4 = 8$ bytes

int arr[3][3]

9 integers \times 4 bytes = 36 bytes.

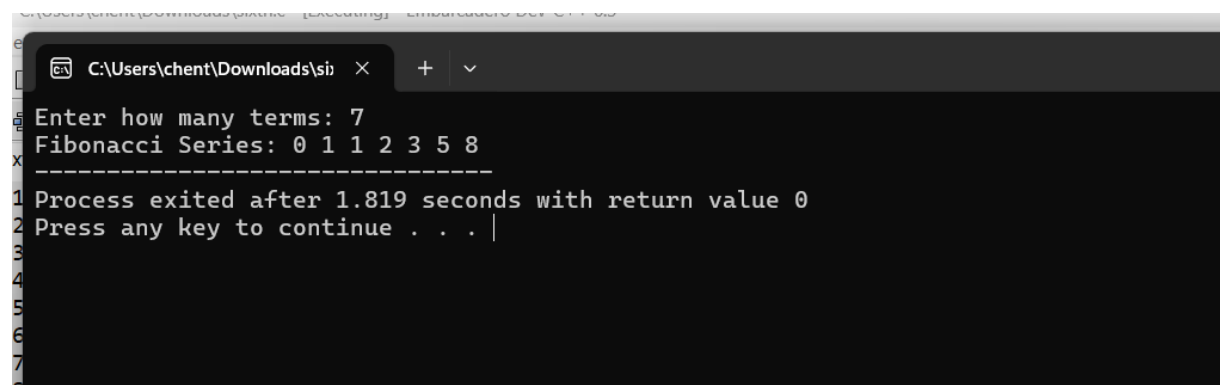
So total space used in the function is $4+4+8+36 = 52$ bytes, which is a constant.

Question 6: Write a program to calculate Fibonacci of a number.

CODE:

```
1  #include <stdio.h>
2
3  int main() {
4      int n, i;
5      int a = 0, b = 1, next;
6
7      printf("Enter how many terms: ");
8      scanf("%d", &n);
9
10     printf("Fibonacci Series: %d %d ", a, b);
11
12     for (i = 2; i < n; i++) {
13         next = a + b;
14         printf("%d ", next);
15         a = b;
16         b = next;
17     }
18
19     return 0;
20 }
21
```

OUTPUT:



```
C:\Users\chent\Downloads\si... x + v
Enter how many terms: 7
Fibonacci Series: 0 1 1 2 3 5 8
-----
1 Process exited after 1.819 seconds with return value 0
2 Press any key to continue . . . |
3
4
5
6
7
8
```

SSSSSSS

Space Complexity: O(1)

In main()

int n → 4 bytes

int i → 4 bytes

int a → 4 bytes

int b → 4 bytes

int c → 4 bytes

So the total space used is 20 bytes which is a constant