

Design Analysis and Algorithm – Lab Work

Week 1

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

Code:

```
#include<stdio.h>
void sumofnum(int n, int s){
    s=0;
    for(int i=1;i<=n;i++){
        s=s+i;
    }
    printf("The sum of n natural numbers are: %d",s);
}
int main(){
    int n;
    printf("Enter the value for n:");
    scanf("%d", &n);
    sumofnum(n,0);
    return 0;
}
```

Output:

```
Enter the value for n:5
The sum of n natural numbers are: 15
-----
```

Space Complexity: $O(1)$ – The above algorithm uses fixed memory which does not depend on n.

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

Code:

```

#include<stdio.h>
void sumofnum(int n, int s){
    s=0;
    for(int i=1;i<=n;i++){
        s=s+i*i;
    }
    printf("The sum of squares of n natural numbers are: %d",s);
}
int main(){
    int n;
    printf("Enter the value for n:");
    scanf("%d", &n);
    sumofnum(n,0);
    return 0;
}

```

Output:

```

Enter the value for n:3
The sum of squares of n natural numbers are: 14

```

Space Complexity: : $O(1)$ – The above algorithm uses fixed memory which does not depend on n.

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

Code:

```

#include<stdio.h>
void sumofnum(int n, int s){
    s=0;
    for(int i=1;i<=n;i++){
        s=s+i*i*i;
    }
    printf("The sum of squares of n natural numbers are: %d",s);
}
int main(){
    int n;
    printf("Enter the value for n:");
    scanf("%d", &n);
    sumofnum(n,0);
    return 0;
}

```

Output:

```

Enter the value for n:3
The sum of squares of n natural numbers are: 36

```

Space Complexity: $O(1)$ – The above algorithm uses fixed memory which does not depend on n .

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

Code:

```
#include <stdio.h>
int fact(int n) {
    if (n == 0 || n == 1)
        return 1;
    return n * fact(n - 1);
}
int main() {
    int n;
    printf("Enter the value for n: ");
    scanf("%d", &n);
    int f = fact(n);
    printf("The factorial of the given number is: %d", f);
    return 0;
}
```

Output:

```
Enter the value for n: 5
The factorial of the given number is: 120
```

Space Complexity: $O(N)$ - Recursive functions use stack memory, which increases with the recursion depth. So the stack grows linearly with n .

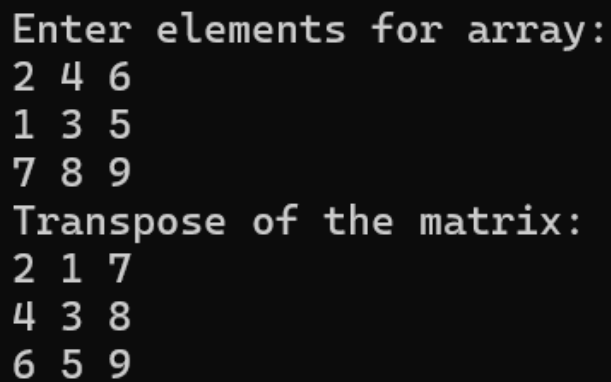
Question 5: Write a program for transposing a 3 x 3 matrix.

Code:

```
#include<stdio.h>
int main() {
    int a[3][3], b[3][3];
    int i, j;
    printf("Enter elements for array:\n");

    for(i=0;i<3;i++) {
        for(j=0;j<3;j++) {
            scanf("%d", &a[i][j]);
        }
    }
    for(i=0;i<3;i++) {
        for(j=0;j<3;j++) {
            b[j][i] = a[i][j];
        }
    }
    printf("Transpose of the matrix:\n");
    for(i=0;i<3;i++) {
        for(j=0;j<3;j++) {
            printf("%d ", b[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

Output:



```
Enter elements for array:
2 4 6
1 3 5
7 8 9
Transpose of the matrix:
2 1 7
4 3 8
6 5 9
```

Space Complexity: $O(1)$,because the size of the matrix is fixed.

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

Code:

```
#include <stdio.h>
int fib(int n){
    int a=0,b=1,c;
    if(n==0)
        return 0;
    for(int i=2;i<=n;i++){
        c=a+b;
        a=b;
        b=c;
    }
    return b;
}
int main(){
    int n;
    printf("Enter n: ");
    scanf("%d", &n);
    printf("Fibonacci number is: %d", fib(n));
    return 0;
}
```

Output:

```
Enter n: 8
Fibonacci number is: 21
```

Space Complexity: $O(1)$ – The above algorithm uses fixed memory which does not depend on n .