

1st WEEK

Name: A. Jeeviteswara Reddy

Roll no:CH.SC.U4CSE24164

Program 1:

Code:

```
#include <stdio.h>
int sum(int n){
    int sum=0;
    for(int i=1;i<=n;i++){
        sum+=i;
    }
    return sum;
}
int main(){
    int n;
    printf("Enter a number: ");
    scanf("%d",&n);
    int a=sum(n);
    printf("Sum of first %d natural numbers = %d\n", n, a);
    return 0;
}
```

Output:

```
C:\Users\A JEEVITESWARA REDDY>sum  
Enter a number: 10  
Sum of first 10 natural numbers = 55
```

Space-Complexity Justification:

Fixed part:

Variables: n,a(in main function),n,sum,i(in the function)

Space used: $(1+1+1+1+1)*4=5*4=20$

Variable part:

No array, recursion or any data structure used, So

Space used: 0

Space = 20= constant

Space complexity = $O(1)$

Program 2:

Code:

```
#include <stdio.h>  
int main(){  
    int n,sum=0;  
    printf("Enter a number: ");  
    scanf("%d",&n);  
    for(int i=1;i<=n;i++){  
        sum+=i * i;  
    }  
    printf("Sum of squares of first %d natural numbers = %d\n", n, sum);  
    return 0;  
}
```

Output:

```
C:\Users\A JEEVITESWARA REDDY>sum2
Enter a number: 10
Sum of squares of first 10 natural numbers = 385
```

Space-Complexity Justification:

Fixed part:

Variables: n,sum,i(in main function)

Space used: $(1+1+1)*4=3*4=12$

Variable part:

No array, recursion or any data structure used, So

Space used: 0

Space = 12= constant

Space complexity = $O(1)$

Program 3:

Code:

```
#include <stdio.h>
int main(){
    int n,sum=0;
    printf("Enter a number: ");
    scanf("%d",&n);
    for(int i=1;i<=n;i++){
        sum+=i * i * i;
    }
    printf("Sum of cubes of first %d natural numbers = %d\n", n, sum);
    return 0;
}
```

Output:

```
C:\Users\A JEEVITESWARA REDDY>sum3
Enter a number: 10
Sum of cubes of first 10 natural numbers = 3025
```

Space-Complexity Justification:

Fixed part:

Variables: n,sum,i(in main function)

Space used: $(1+1+1)*4=3*4=12$

Variable part:

No array, recursion or any data structure used, So

Space used: 0

Space = 12 = constant

Space complexity = $O(1)$

Program 4:

Code:

```

#include <stdio.h>
int factorial(int n){
    if(n==0||n==1)
        return 1;
    return n*factorial(n-1);
}
int main(){
    int n;
    printf("Enter a number: ");
    scanf("%d",&n);
    printf("Factorial of %d = %d\n", n, factorial(n));
    return 0;
}

```

Output:

```

C:\Users\A JEEVITESWARA REDDY>sum4
Enter a number:
5
Factorial of 5 = 120

```

Space-Complexity Justification:

Fixed part:

Variables: n(in main function),n(in the function)

Space used: $(1+1)*4=2*4=8$

Variable part:

Fact(n) calls Fact(n-1) until n becomes 0 or 1

Recursion for n times, So

Space used: n

Space = $8+n$

Space complexity = $O(n)$

Program 5:

Code:

```
#include <stdio.h>
int main(){
    int r, c;
    printf("Enter rows and columns: ");
    scanf("%d %d",&r,&c);
    int a[3][3],t[3][3];
    printf("Enter matrix elements:\n");
    for(int i=0;i<r;i++){
        for(int j=0;j<c;j++){
            scanf("%d",&a[i][j]);
        }
    }
    for(int i=0;i<r;i++){
        for(int j=0;j<c;j++){
            t[j][i]=a[i][j];
        }
    }
    printf("Transpose of the matrix:\n");
    for(int i=0;i<c;i++){
        for(int j=0;j<r;j++){
            printf("%d",t[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

Output:

```
Enter rows and columns: 3 3
Enter matrix elements:
1 2 3
4 5 6
7 8 9
Transpose of the matrix:
147
258
369
C:\Users\A_JEEVITESWARA REDDY>
```

Space-Complexity Justification:

Fixed part:

Variables: a[3][3], t[3][3], r,c,l,j

Space used: $(3*3+3*3+1+1+1+1)*4 = 22*4 = 88$

Variable part:

Array used in the program doesn't depend on the input n, So

Space used: 0

Space = 88= constant

Space complexity =O(1)

Program 6:

Code:

```
#include <stdio.h>
int main() {
    int n,a=0,b=1,c=0,i=1;
    printf("Enter the number of terms: ");
    scanf("%d", &n);
    printf("Fibonacci sequence: ");
    while(i<=n) {
        printf("%d",a);
        c=a+b;
        a=b;
        b=c;
        i++;
    }
    return 0;
}
```

Output:

```
C:\Users\A JEEVITESWARA REDDY>sum6
Enter the number of terms: 5
Fibonacci sequence: 01123
```

Space-Complexity Justification:

Fixed part:

Variables: n,a,b,c,i(in main function)

Space used: $(1+1+1+1+1)*4 = 5*4 = 20$

Variable part:

Array used in the program doesn't depend on the input
n, So

Space used: 0

Space = 20

Space complexity = $O(1)$