AMRITA VISHWA VIDYAPEETHAM

DESIGN AND ANALYSIS OF ALGORITHMS

LAB WORKBOOK

WEEK - 1

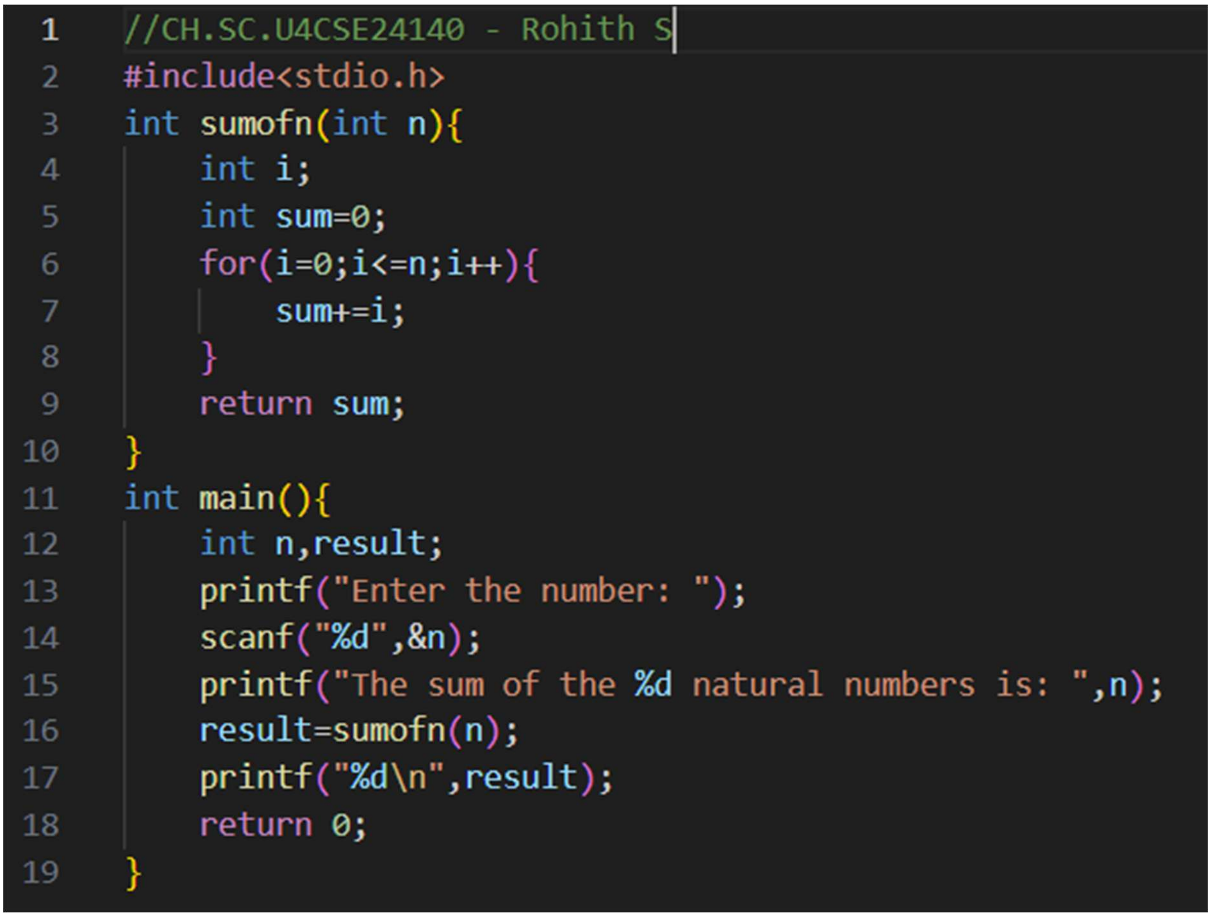
NAME : ROHITH S

ROLL NUMBER : CH.SC.U4CSE24140

CLASS : CSE-B

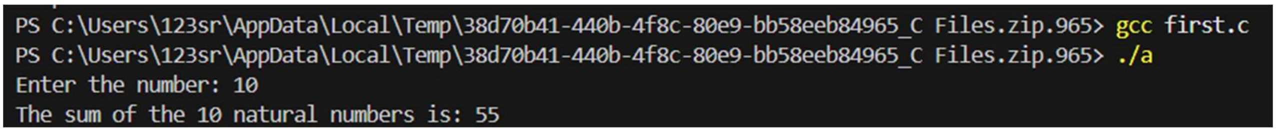
Question 1: Write a program to find sum of first n natural numbers using user defined function.

CODE:



```c
//CH.SC.U4CSE24140 - Rohith S
#include<stdio.h>
int sumofn(int n){
    int i;
    int sum=0;
    for(i=0;i<=n;i++){
        sum+=i;
    }
    return sum;
}
int main(){
    int n,result;
    printf("Enter the number: ");
    scanf("%d",&n);
    printf("The sum of the %d natural numbers is: ",n);
    result=sumofn(n);
    printf("%d\n",result);
    return 0;
}
```

OUTPUT:



```
PS C:\Users\123sr\AppData\Local\Temp\38d70b41-440b-4f8c-80e9-bb58eeb84965_C Files.zip.965> gcc first.c
PS C:\Users\123sr\AppData\Local\Temp\38d70b41-440b-4f8c-80e9-bb58eeb84965_C Files.zip.965> ./a
Enter the number: 10
The sum of the 10 natural numbers is: 55
```

Space Complexity: O(1)

Variables in main():

- int n → 4 bytes
- int result → 4 bytes

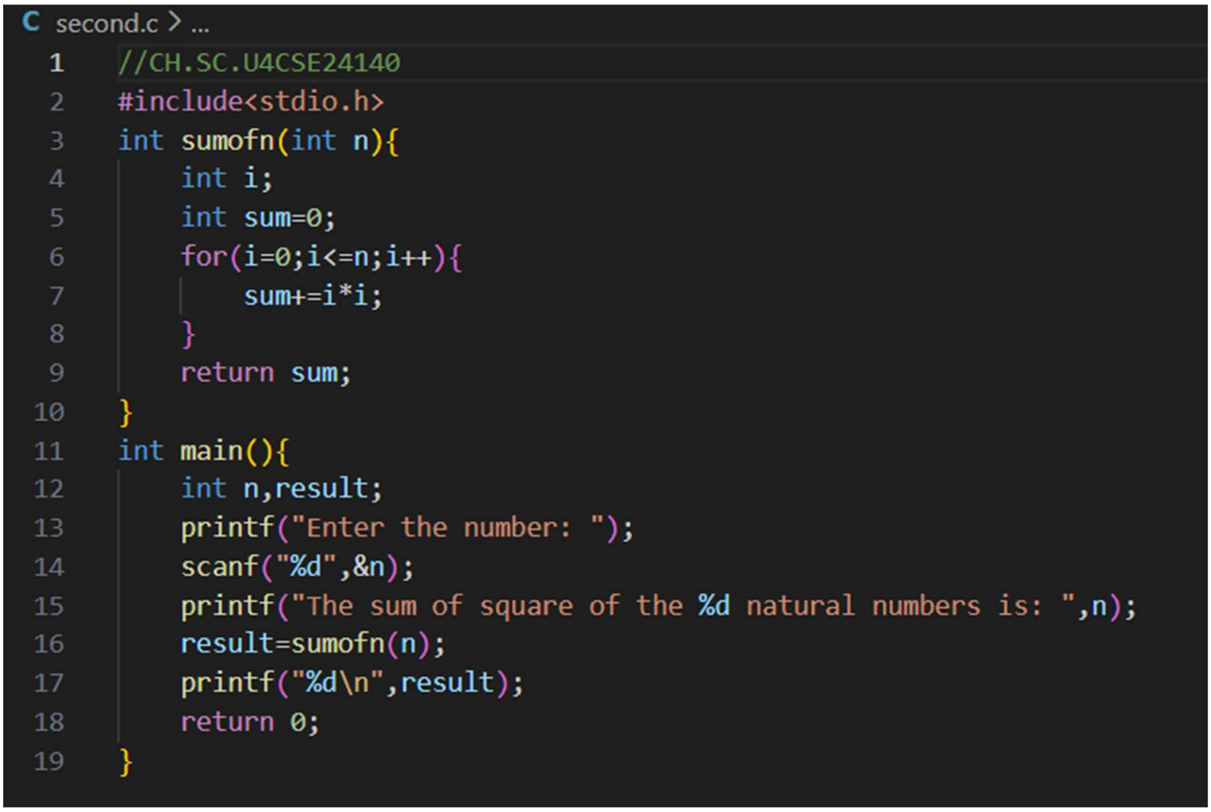
Variables in sumofn():

- int i → 4 bytes
- int sum → 4 bytes

Total memory used by variables:

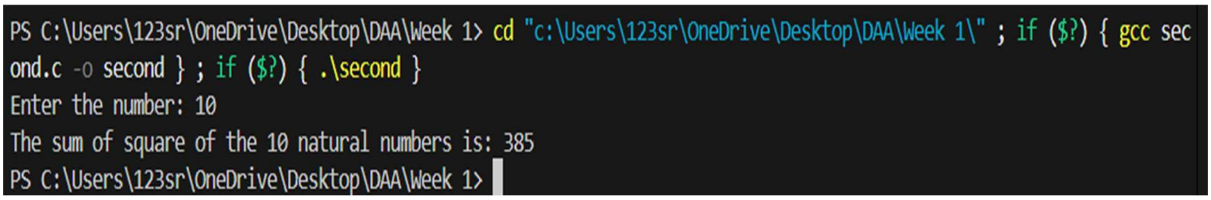
4 + 4 + 4 + 4 = 16 bytes

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



```c
//CH.SC.U4CSE24140
#include<stdio.h>
int sumofn(int n){
    int i;
    int sum=0;
    for(i=0;i<=n;i++){
        sum+=i*i;
    }
    return sum;
}
int main(){
    int n,result;
    printf("Enter the number: ");
    scanf("%d",&n);
    printf("The sum of square of the %d natural numbers is: ",n);
    result=sumofn(n);
    printf("%d\n",result);
    return 0;
}
```

OUTPUT:



```
PS C:\Users\123sr\OneDrive\Desktop\DAA\Week 1> cd "c:\Users\123sr\OneDrive\Desktop\DAA\Week 1\" ; if ($?) { gcc sec
ond.c -o second } ; if ($?) { .\second }
Enter the number: 10
The sum of square of the 10 natural numbers is: 385
PS C:\Users\123sr\OneDrive\Desktop\DAA\Week 1>
```

Space Complexity: O(1)

Variables in main():

- int n → 4 bytes
- int result → 4 bytes

Variables in sumofn():

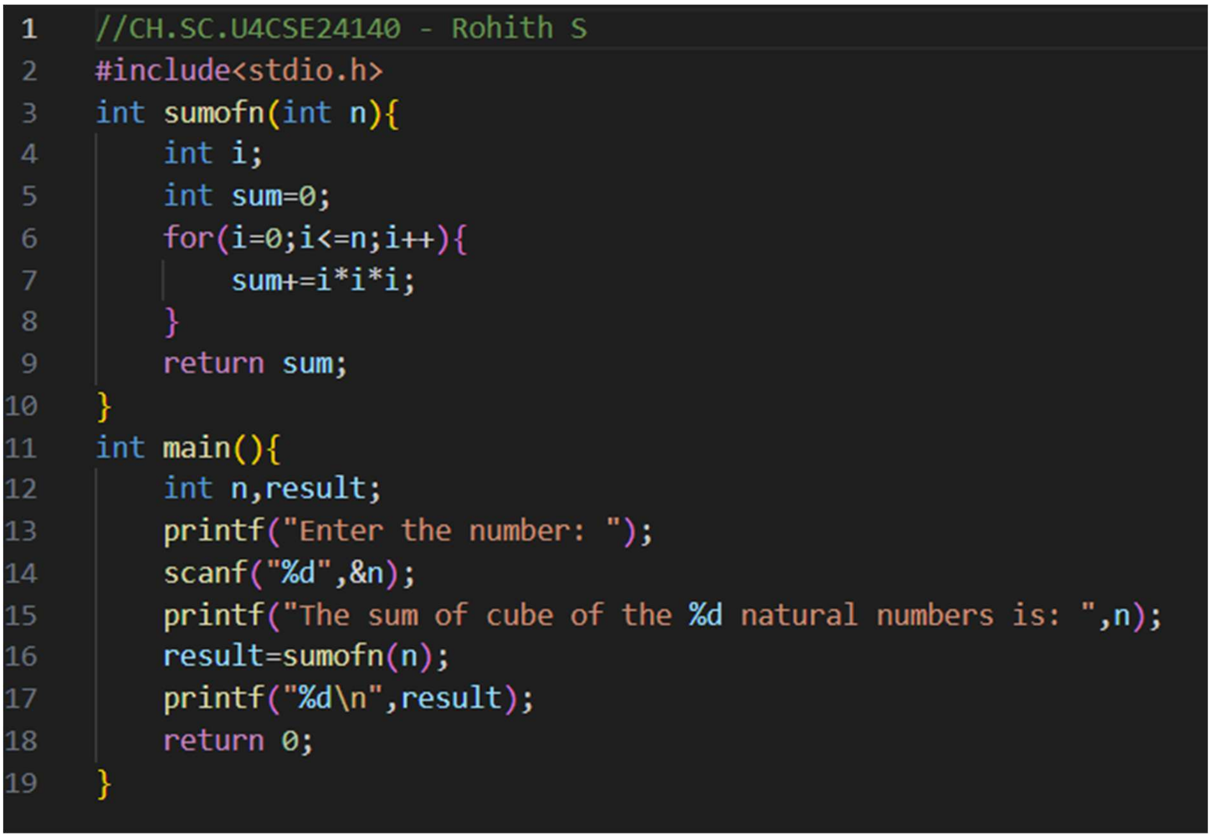
- int i → 4 bytes
- int sum → 4 bytes
- int n (parameter) → 4 bytes

Total memory used by variables:

4 + 4 + 4 + 4 + 4 = 20 bytes

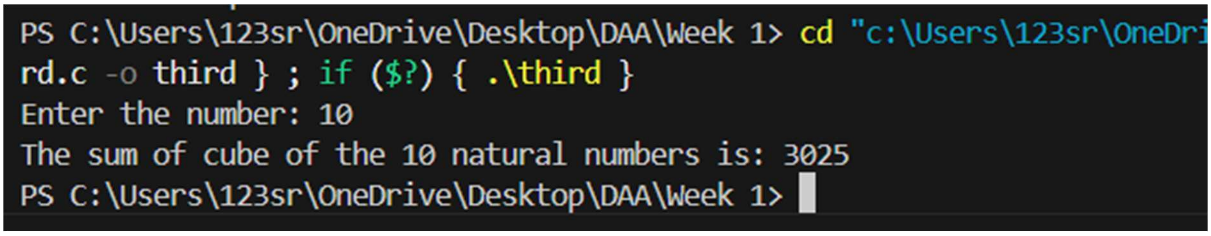
Question 3: Write a program to find sum of cubes of the first n natural numbers.

CODE:



```c
//CH.SC.U4CSE24140 - Rohith S
#include<stdio.h>
int sumofn(int n){
    int i;
    int sum=0;
    for(i=0;i<=n;i++){
        sum+=i*i*i;
    }
    return sum;
}
int main(){
    int n,result;
    printf("Enter the number: ");
    scanf("%d",&n);
    printf("The sum of cube of the %d natural numbers is: ",n);
    result=sumofn(n);
    printf("%d\n",result);
    return 0;
}
```

OUTPUT:



```
PS C:\Users\123sr\OneDrive\Desktop\DAA\Week 1> cd "c:\Users\123sr\OneDri
rd.c -o third } ; if ($?) { .\third }
Enter the number: 10
The sum of cube of the 10 natural numbers is: 3025
PS C:\Users\123sr\OneDrive\Desktop\DAA\Week 1> 
```

Space Complexity: O(1)

Variables in main():

- int n → 4 bytes
- int result → 4 bytes

Variables in sumofn():

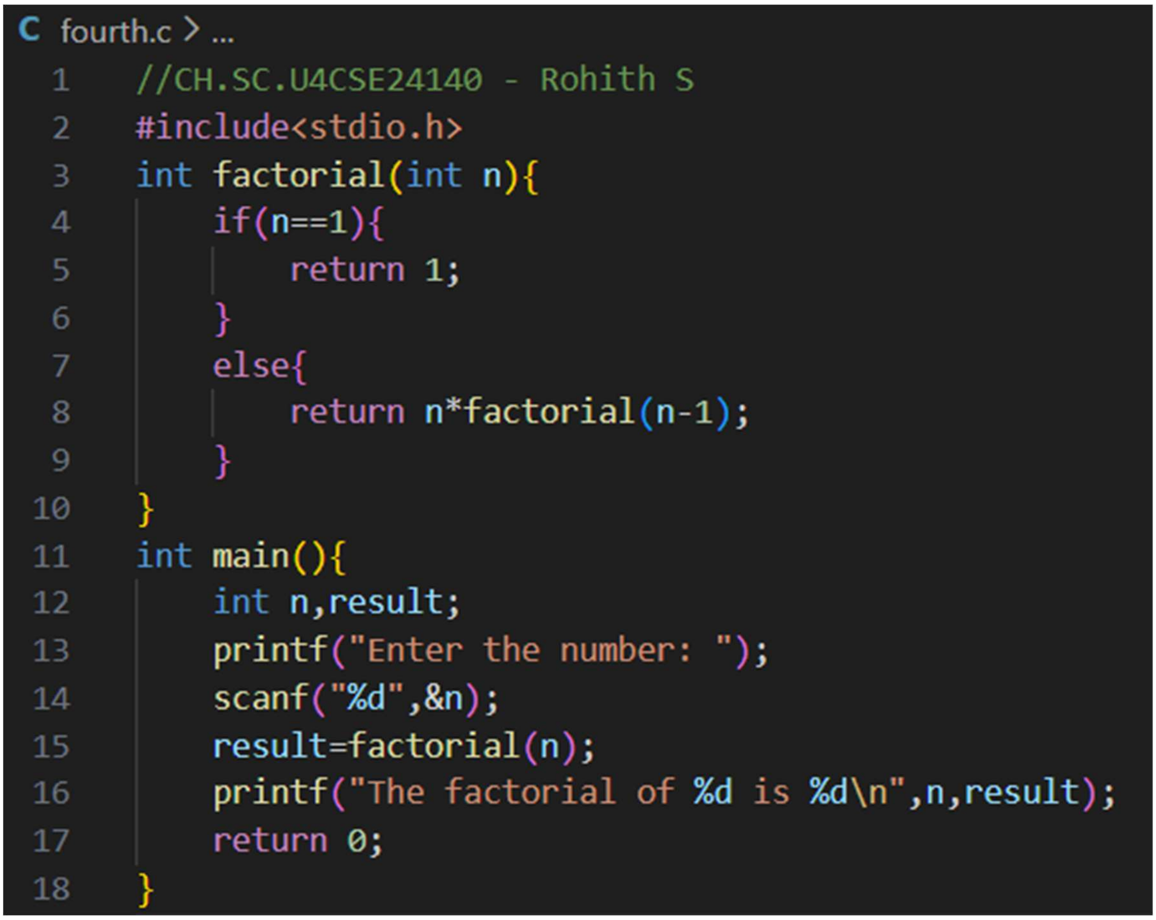
- int n (parameter) → 4 bytes
- int i → 4 bytes
- int sum → 4 bytes

Total memory used by variables:

4 + 4 + 4 + 4 + 4 = 20 bytes

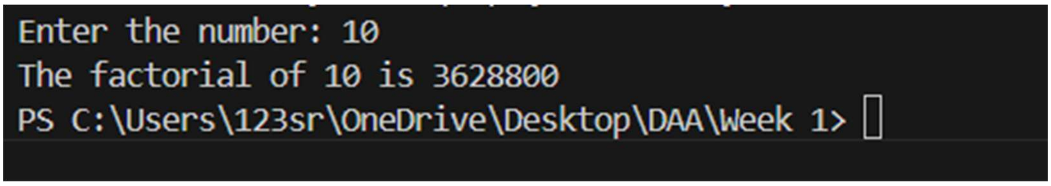
Question 4: Write a program to find the factorial of a given integer using recursion.

CODE:



```c
//CH.SC.U4CSE24140 - Rohith S
#include<stdio.h>
int factorial(int n){
    if(n==1){
        return 1;
    }
    else{
        return n*factorial(n-1);
    }
}
int main(){
    int n,result;
    printf("Enter the number: ");
    scanf("%d",&n);
    result=factorial(n);
    printf("The factorial of %d is %d\n",n,result);
    return 0;
}
```

OUTPUT:



```
Enter the number: 10
The factorial of 10 is 3628800
PS C:\Users\123sr\OneDrive\Desktop\DAA\Week 1> 
```

Space Complexity: O(n)

Variables in main():

- int n → 4 bytes
- int result → 4 bytes

Variables in factorial() (per call):

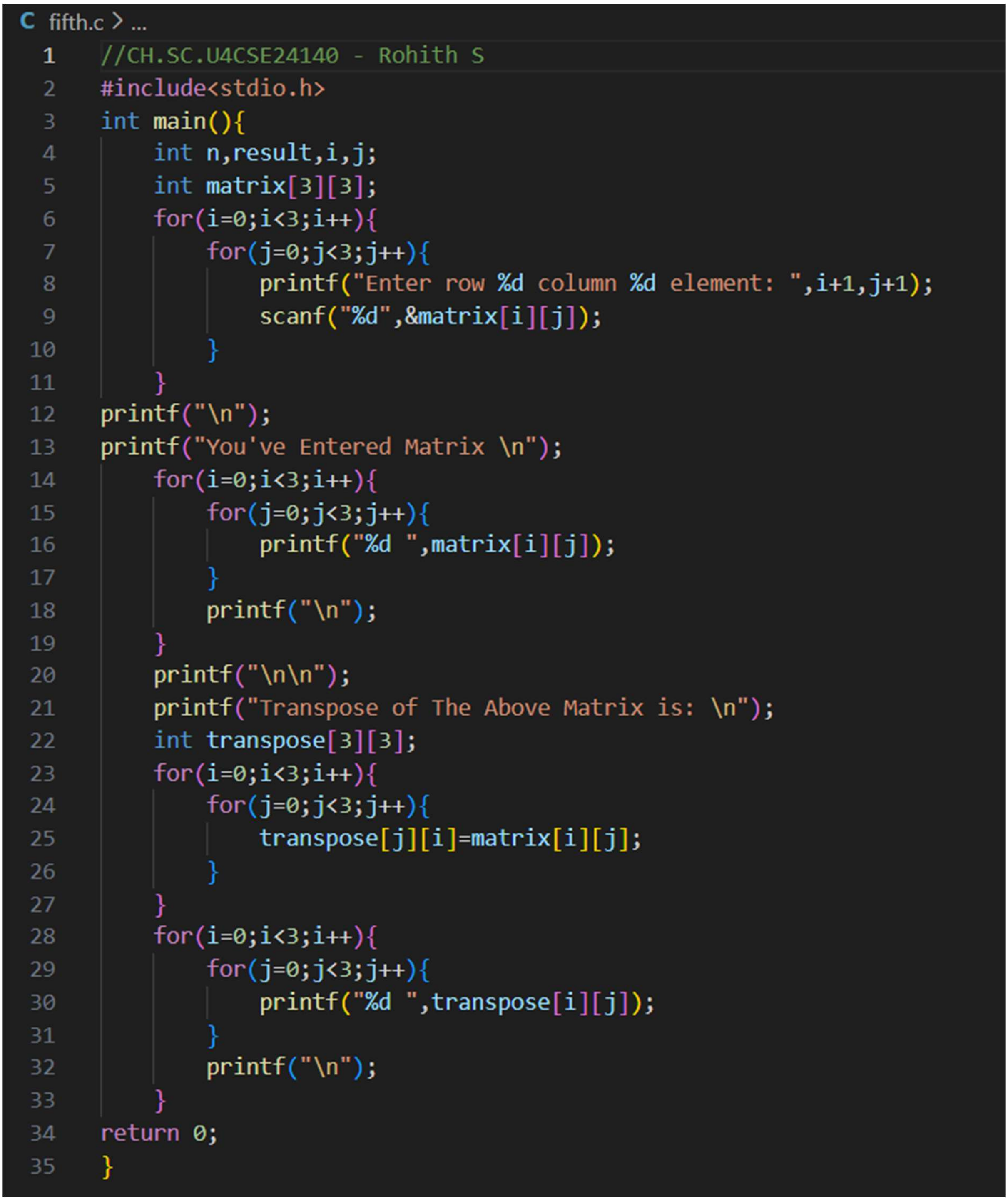
- int n (parameter) → 4 bytes

Total memory used by recursion stack:

4 × n byte = O(n)

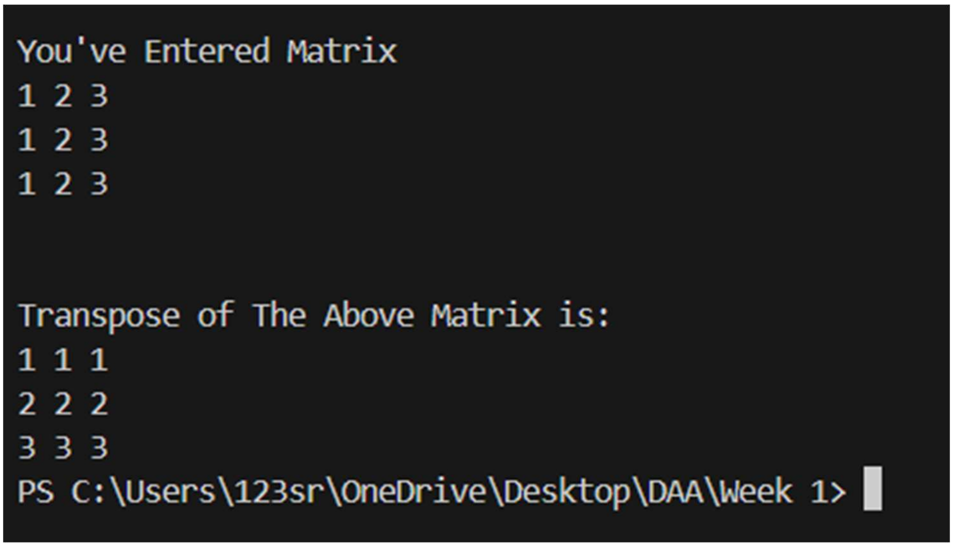
Question 5: Write a program for transposing a 3 x 3 matrix.

CODE:



```c
//CH.SC.U4CSE24140 - Rohith S
#include<stdio.h>
int main(){
    int n,result,i,j;
    int matrix[3][3];
    for(i=0;i<3;i++){
        for(j=0;j<3;j++){
            printf("Enter row %d column %d element: ",i+1,j+1);
            scanf("%d",&matrix[i][j]);
        }
    }
printf("\n");
printf("You've Entered Matrix \n");
    for(i=0;i<3;i++){
        for(j=0;j<3;j++){
            printf("%d ",matrix[i][j]);
        }
        printf("\n");
    }
    printf("\n\n");
    printf("Transpose of The Above Matrix is: \n");
    int transpose[3][3];
    for(i=0;i<3;i++){
        for(j=0;j<3;j++){
            transpose[j][i]=matrix[i][j];
        }
    }
    for(i=0;i<3;i++){
        for(j=0;j<3;j++){
            printf("%d ",transpose[i][j]);
        }
        printf("\n");
    }
return 0;
}
```

OUTPUT:



```
You've Entered Matrix
1 2 3
1 2 3
1 2 3


Transpose of The Above Matrix is:
1 1 1
2 2 2
3 3 3
PS C:\Users\123sr\OneDrive\Desktop\DAA\Week 1>
```

Space Complexity: O(1)

Variables in main():

- int n → 4 bytes
- int result → 4 bytes
- int i → 4 bytes
- int j → 4 bytes

Arrays in main():

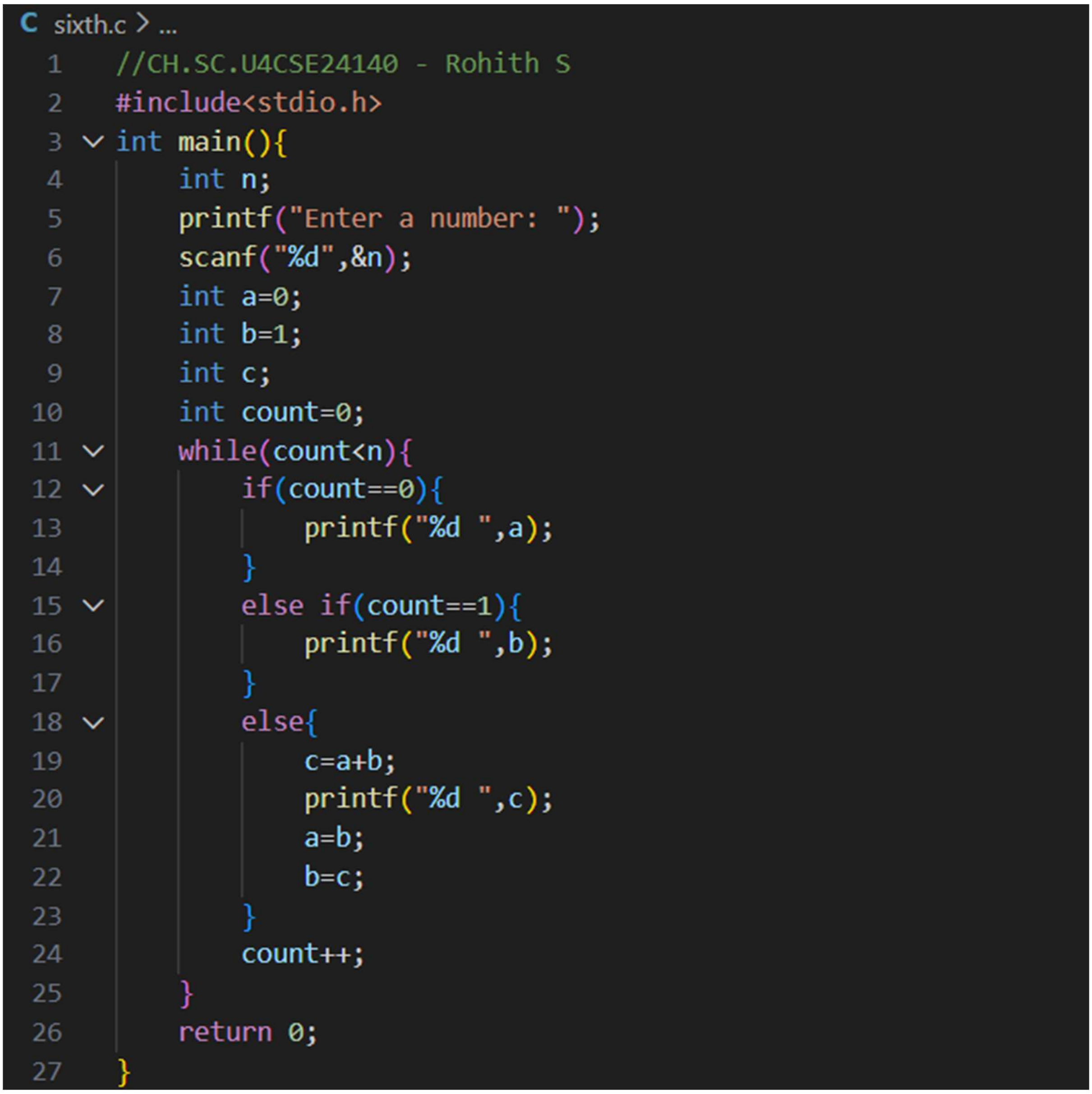
- int matrix[3][3] → 9 × 4 = 36 bytes
- int transpose[3][3] → 9 × 4 = 36 bytes

Total memory used by variables and arrays:

4 + 4 + 4 + 4 + 36 + 36 = 88 bytes

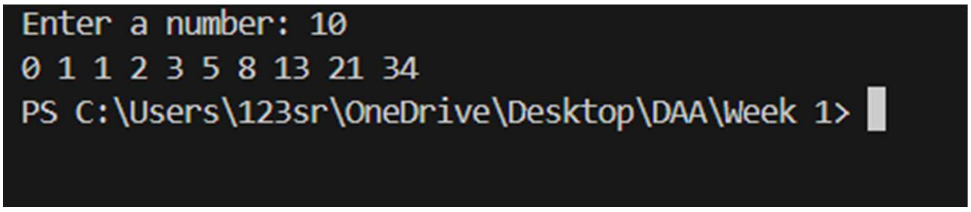
Question 6: Write a program to calculate Fibonacci of a number.

CODE:



```c
//CH.SC.U4CSE24140 - Rohith S
#include<stdio.h>
int main(){
    int n;
    printf("Enter a number: ");
    scanf("%d",&n);
    int a=0;
    int b=1;
    int c;
    int count=0;
    while(count<n){
        if(count==0){
            printf("%d ",a);
        }
        else if(count==1){
            printf("%d ",b);
        }
        else{
            c=a+b;
            printf("%d ",c);
            a=b;
            b=c;
        }
        count++;
    }
    return 0;
}
```

OUTPUT:



```
Enter a number: 10
0 1 1 2 3 5 8 13 21 34
PS C:\Users\123sr\OneDrive\Desktop\DAA\Week 1>
```

Space Complexity: O(1)

Variables in main():

- int n → 4 bytes
- int a → 4 bytes
- int b → 4 bytes
- int c → 4 bytes
- int count → 4 bytes

Total memory used by variables:

4 + 4 + 4 + 4 + 4 = 20 bytes