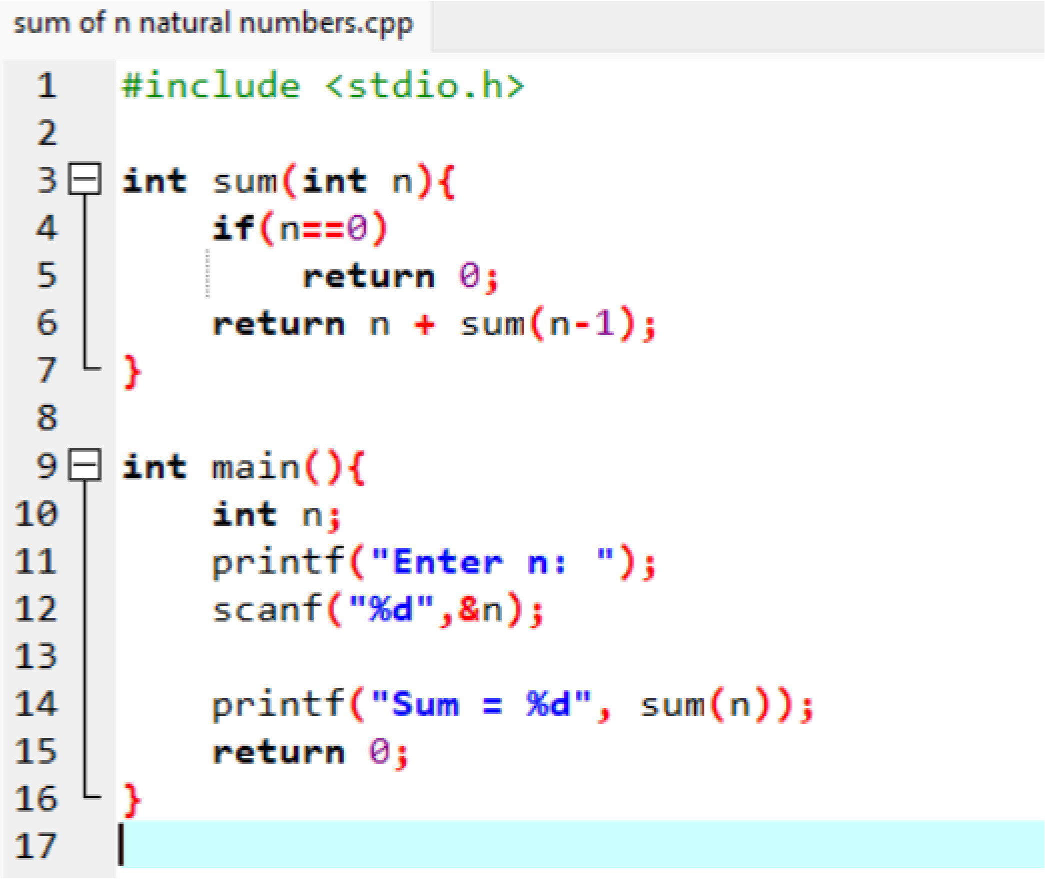
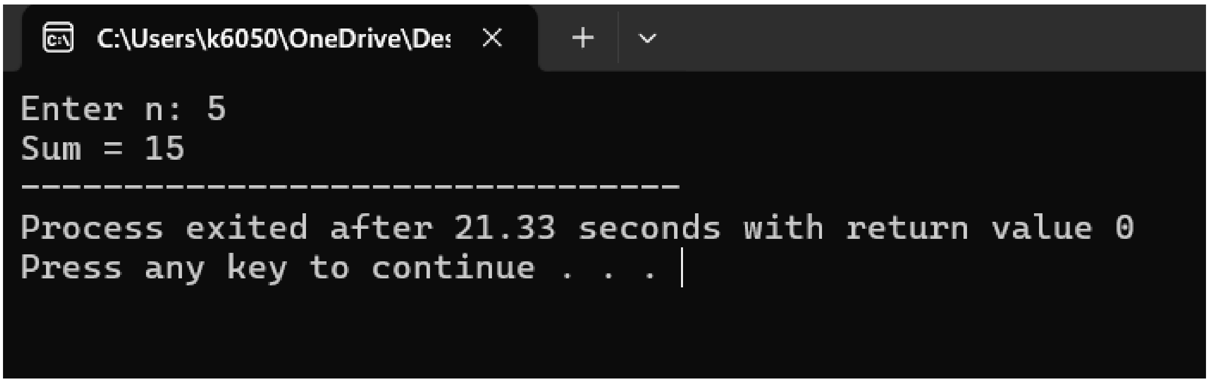
1) Write a program to find the sum of n natural using a user defined function

Code:



```c
#include <stdio.h>

int sum(int n){
	if(n==0)
		return 0;
	return n + sum(n-1);
}

int main(){
	int n;
	printf("Enter n: ");
	scanf("%d",&n);
	
	printf("Sum = %d", sum(n));
	return 0;
}
```

Output:

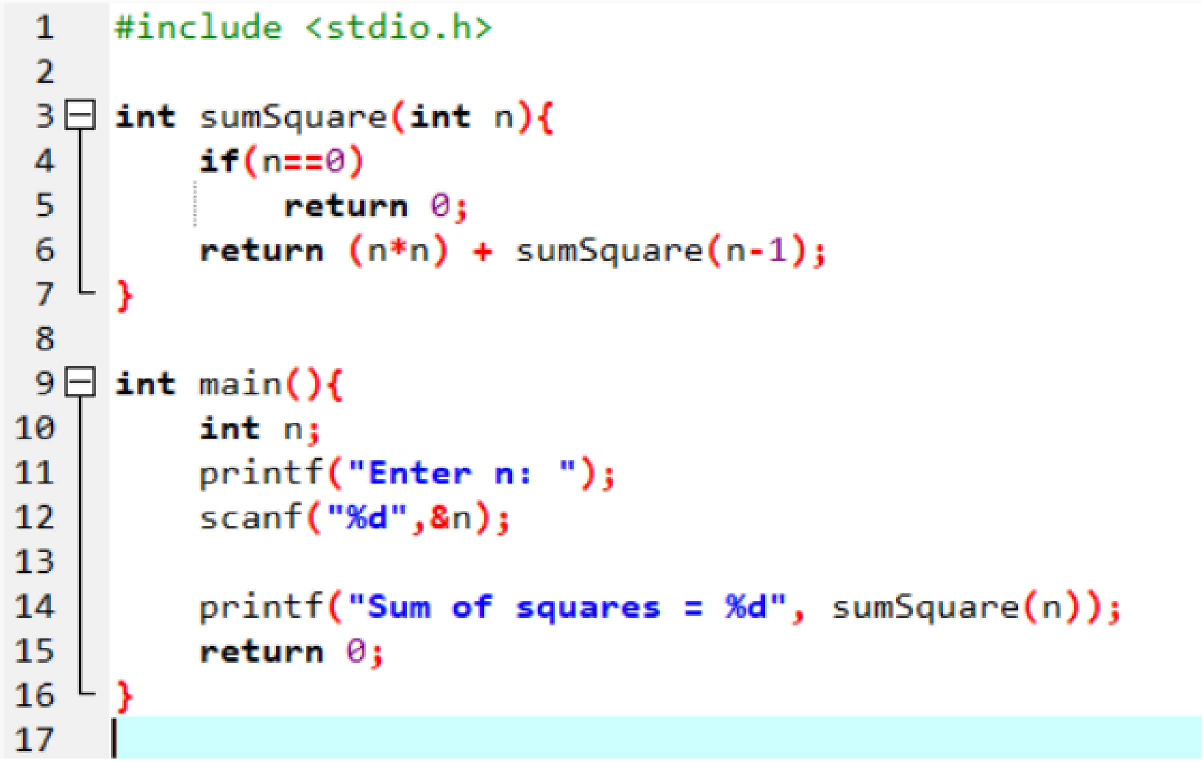


```
Enter n: 5
Sum = 15
--------------------------------
Process exited after 21.33 seconds with return value 0
Press any key to continue . . .
```

This code is telling us the sum of n numbers in recursion and one by one process of implementation till the condition fail.

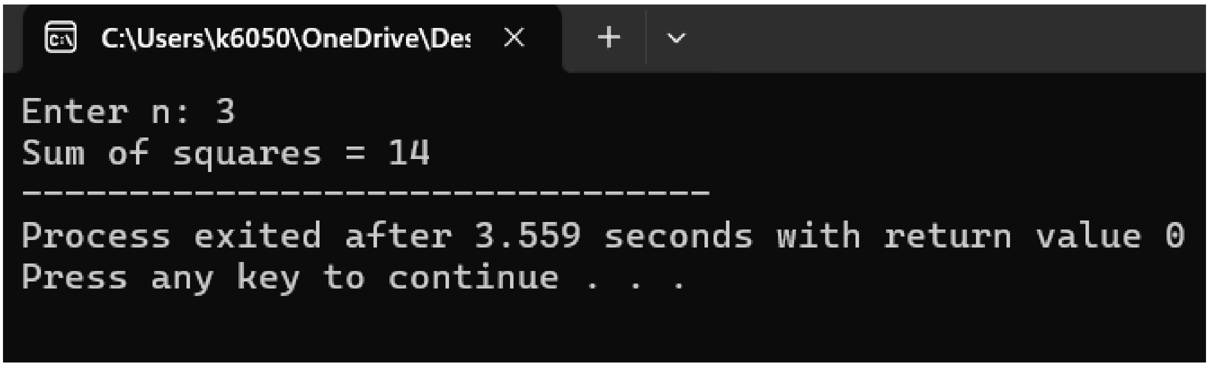
2)Write a program to find the sum of squares of n natural numbers

Code:



```c
#include <stdio.h>

int sumSquare(int n){
	if(n==0)
		return 0;
	return (n*n) + sumSquare(n-1);
}

int main(){
	int n;
	printf("Enter n: ");
	scanf("%d",&n);

	printf("Sum of squares = %d", sumSquare(n));
	return 0;
}
```

OUTPUT:

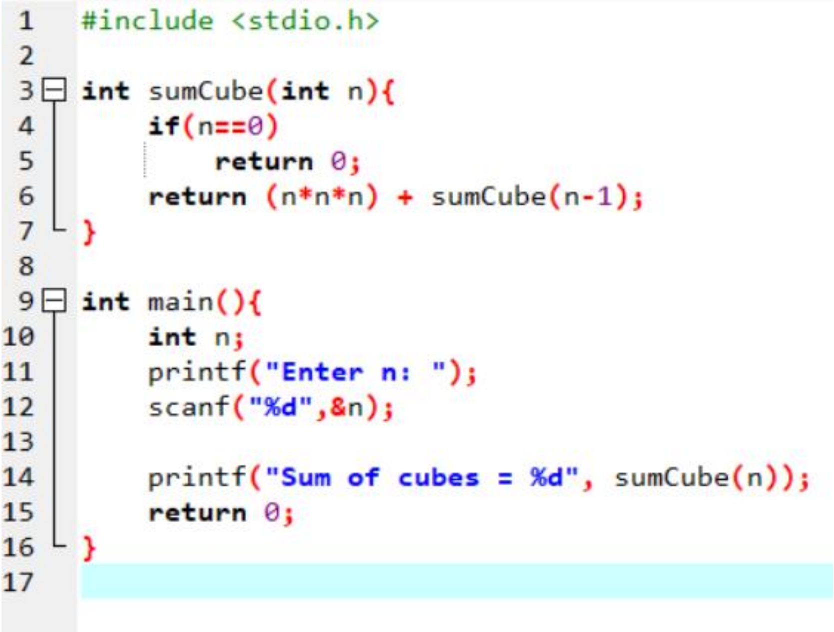


```
Enter n: 3
Sum of squares = 14
--------------------------------
Process exited after 3.559 seconds with return value 0
Press any key to continue . . .
```

This program uses recursion to calculate the sum of squares of the first *n* natural numbers.
It repeatedly calls itself with decreasing values of *n* until reaching the base case (n = 0)

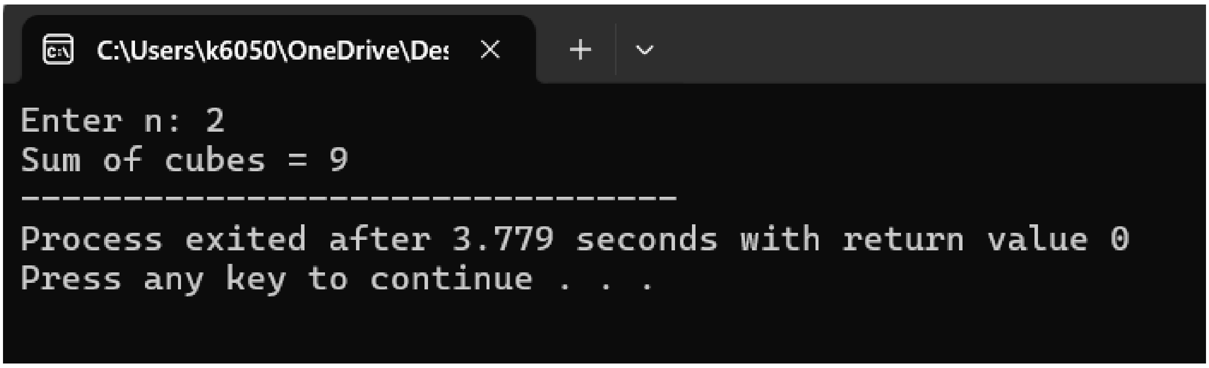
3)Write a program to find the sum of cubes of n natural the sum of cubes of n natural numbers

CODE:



```c
#include <stdio.h>

int sumCube(int n){
    if(n==0)
        return 0;
    return (n*n*n) + sumCube(n-1);
}

int main(){
    int n;
    printf("Enter n: ");
    scanf("%d",&n);

    printf("Sum of cubes = %d", sumCube(n));
    return 0;
}

```

OUTPUT:



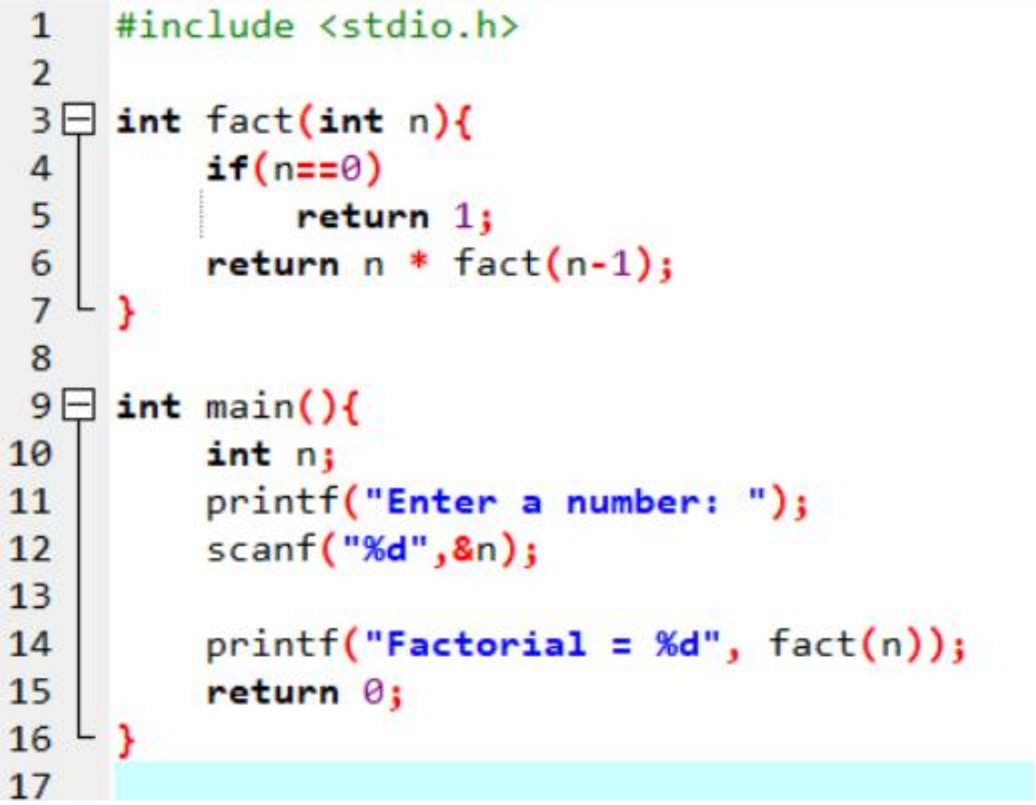
```
Enter n: 2
Sum of cubes = 9
--------------------------------
Process exited after 3.779 seconds with return value 0
Press any key to continue . . .
```

.

This program adds the cube of each number from *n* to 1 using recursive calls. When *n* = *0*, the recursion stops and returns the total sum of cubes.

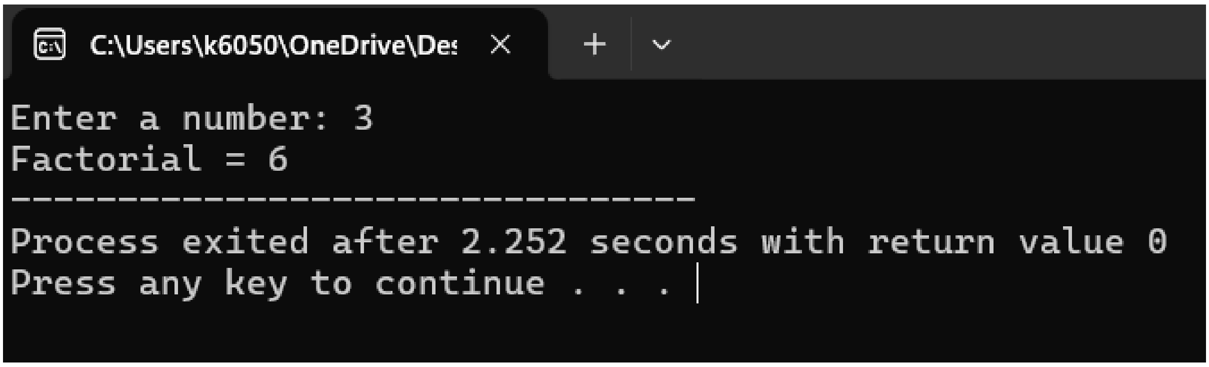
4)Write a program to find the factorial of a given number using recursion

CODE:



```c
#include <stdio.h>

int fact(int n){
	if(n==0)
		return 1;
	return n * fact(n-1);
}

int main(){
	int n;
	printf("Enter a number: ");
	scanf("%d",&n);
	
	printf("Factorial = %d", fact(n));
	return 0;
}

```

OUTPUT:



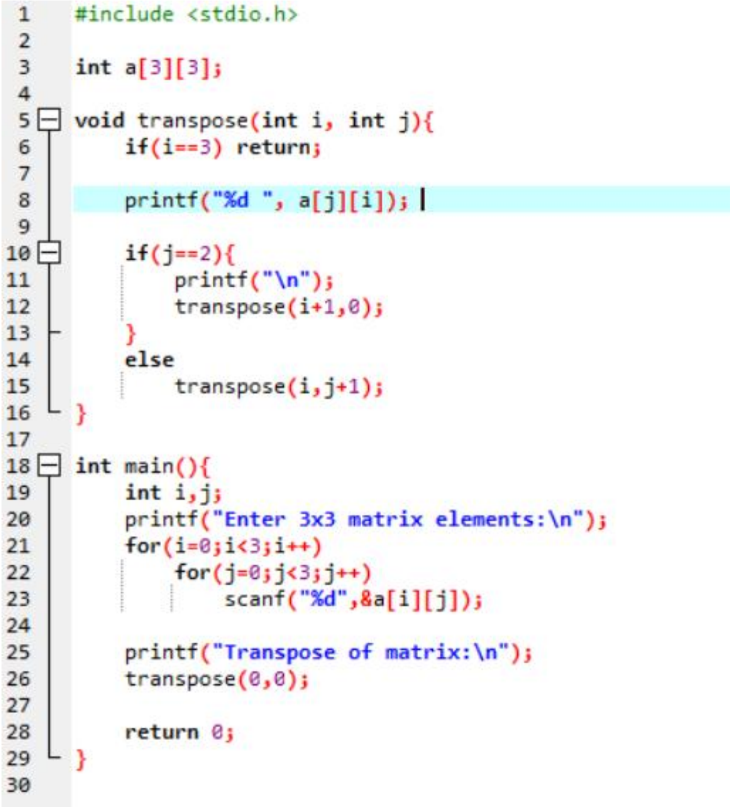
```
Enter a number: 3
Factorial = 6
--------------------------------
Process exited after 2.252 seconds with return value 0
Press any key to continue . . .
```

The program multiplies numbers from *n* down to 1 using recursive function calls.

It ends when the base case *n = 0* returns 1, completing the factorial calculation.

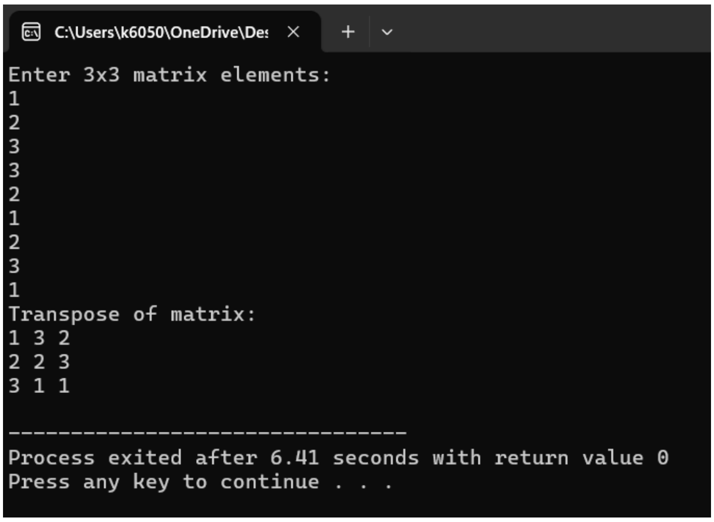
5)Write a program to transpose a 3x3 matrix

CODE:



```c
#include <stdio.h>

int a[3][3];

void transpose(int i, int j){
    if(i==3) return;

    printf("%d ", a[j][i]);

    if(j==2){
        printf("\n");
        transpose(i+1,0);
    }
    else
        transpose(i,j+1);
}

int main(){
    int i,j;
    printf("Enter 3x3 matrix elements:\n");
    for(i=0;i<3;i++)
        for(j=0;j<3;j++)
            scanf("%d",&a[i][j]);

    printf("Transpose of matrix:\n");
    transpose(0,0);

    return 0;
}
```

OUTPUT:



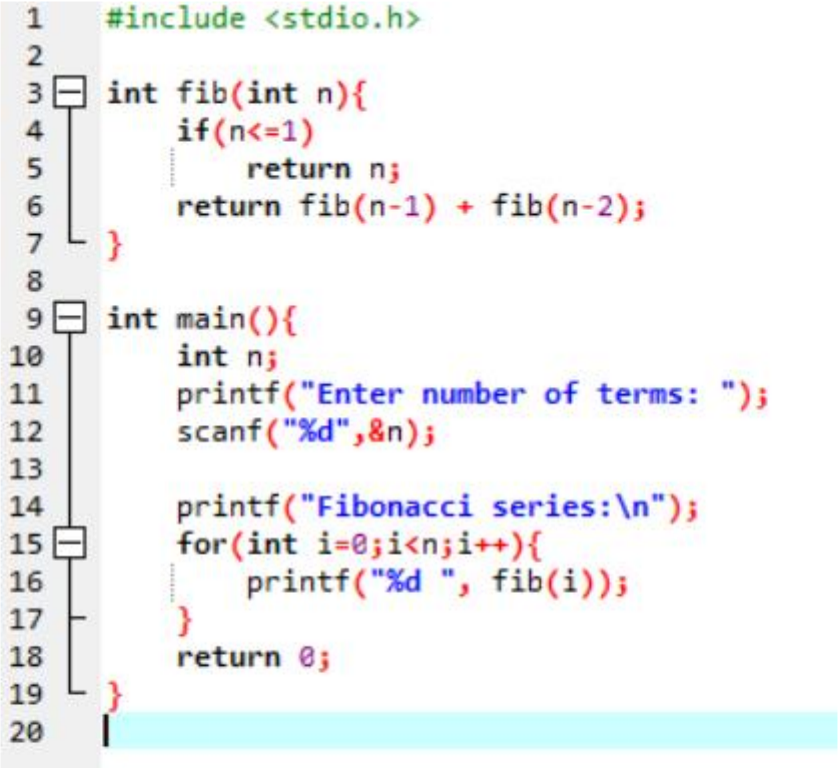
```
Enter 3x3 matrix elements:
1
2
3
3
2
1
2
3
1
Transpose of matrix:
1 3 2
2 2 3
3 1 1

--------------------------------
Process exited after 6.41 seconds with return value 0
Press any key to continue . . .
```

The recursion visits each element in row–column order and prints the transposed element (a[j][i]).

It terminates when all rows and columns have been processed.

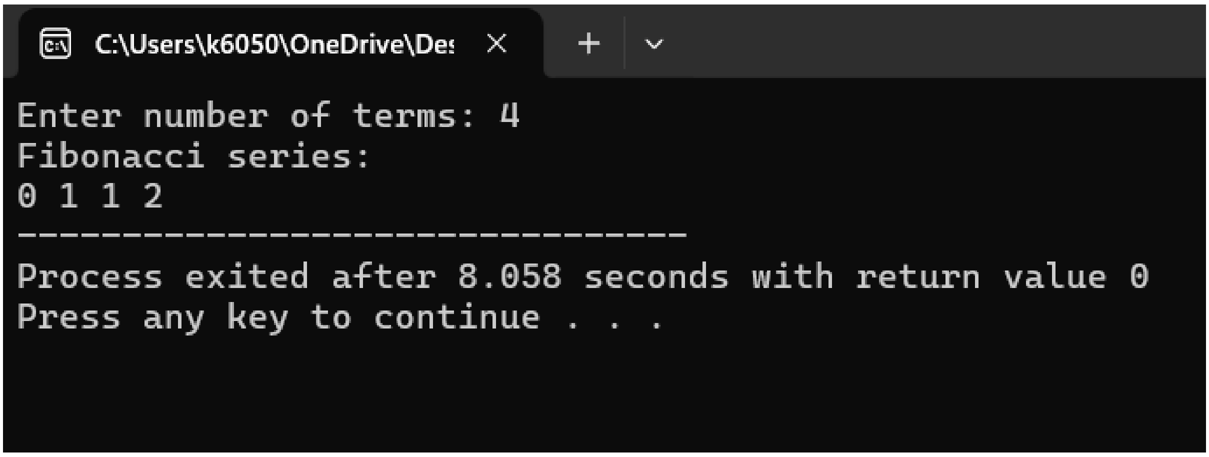
6) Write a program to find the Fibonacci series

CODE:



```c
#include <stdio.h>

int fib(int n){
    if(n<=1)
        return n;
    return fib(n-1) + fib(n-2);
}

int main(){
    int n;
    printf("Enter number of terms: ");
    scanf("%d",&n);

    printf("Fibonacci series:\n");
    for(int i=0;i<n;i++){
        printf("%d ", fib(i));
    }
    return 0;
}
```

OUTPUT:



```
Enter number of terms: 4
Fibonacci series:
0 1 1 2
--------------------------------
Process exited after 8.058 seconds with return value 0
Press any key to continue . . .
```

The program prints Fibonacci numbers by recursively calling fib(n−1) and fib(n−2).
The recursion ends when the base cases (0 and 1) return, generating the full series.