```
// RECURSION PART 1

// Recursions - when a function calls itself directly/indirectly.

/*
    Recursion comprises

    1] Base Case (Mandatory) and return mandatory
    2] Recursive Call (Mandatory)
    3] Processings (Optional)

if base case is not include in the code segmentation error occur.

*/

#include<iostream>
using namespace std;

int fact(int n)
{
        //Base Case
        if(n==0 || n==1){
                return 1;
        }
        //Recursive Call
        int recAns = fact(n-1);
        //Processings
        int findAns = n*recAns;
        return findAns;
}

int main()
{
        cout << fact(5) << endl;
}



//-------------------------------------------------------
// Printing n to 1

#include<iostream>
using namespace std;

void print(int n)
{
        //Base Case
        if(n == 0){
                return ;
        }
        //Processings
        cout << n << " ";
        //Recursive call
        print(n-1);

}

int main()
{
```

```
        print(50);
}

//-------------------------------------------------------

// TAIL AND HEAD RECURSION

/* If recursive relation comes after processing,then it is
   tail recursion.

   If recursive relation comes before processing, it is head
   recursion.
*/

//-------------------------------------------------------
// Calculate n to the power m

#include<iostream>
using namespace std;

int power(int n , int m)
{
    if(m==0)
    {
        return 1; // Base case
        }
        int pow = power(n,m-1); // Recursive relation
        return (n*pow);
}

int main()
{
        cout << power(5,3) << endl;
}

//-------------------------------------------------------

// FIBONACCI SERIES --->

#include<iostream>
using namespace std;
int fib(int n)
{
        //Base Case
        if(n == 0 || n == 1)
        {
                return n;
        }
        //Recursive call
        int ans = fib(n-1) + fib(n-2);
        return ans;
}
int main()
{
        cout << fib(8) << endl;
}

//-------------------------------------------------------
```

```
// FIND SUM -->

#include<iostream>
using namespace std;
int Sum(int n)
{
	//Base Case
	if(n == 0)
	{
		return 0;
	}
	//Recursive call
	int ans = n + Sum(n-1);
	return ans;
}
int main()
{
	cout << Sum(5) << endl;
}

//-------------------------------------------------------
```