

What is Recursion?

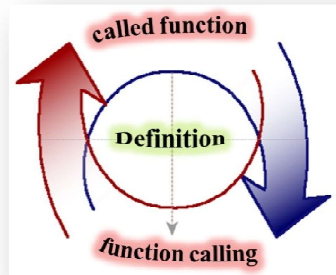
Recursion is a programming technique that allows the programmer to express operations in terms of themselves. In C, this takes the form of a function that calls itself. A useful way to think of recursive functions is to imagine them as a process being performed where one of the instructions is to "repeat the process".

□ Either directly.

- X calls X.

□ Cyclically in a chain.

- X calls Y, and Y calls X.



Used for repetitive computations in which each action is stated in terms of a previous result.

For a problem to be written in recursive form, two conditions are to be satisfied:

- It should be possible to express the problem in recursive form.
- The problem statement must include a stopping condition

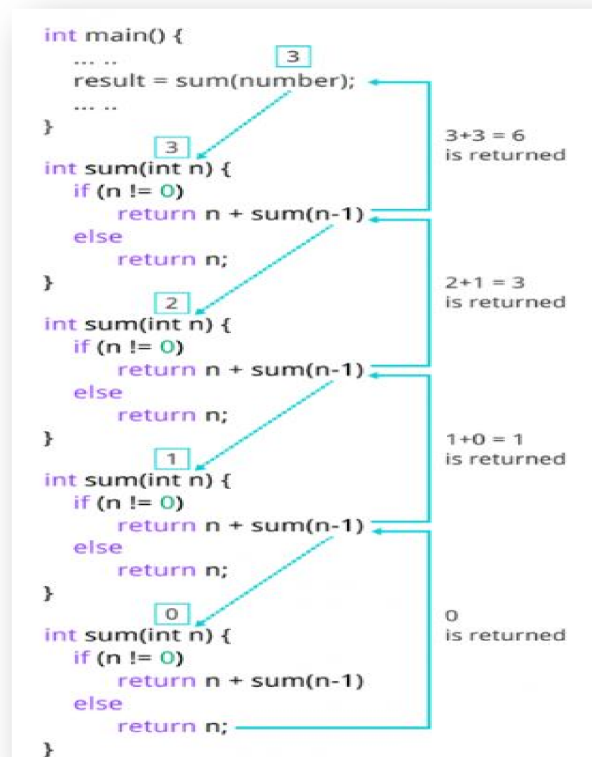
□ Mechanism of Execution:

- When a recursive program is executed, the recursive function calls are not executed immediately.
- They are kept aside (on a stack) until the stopping condition is encountered.
- The function calls are then executed in reverse order.

Example 1: Sum of Natural Numbers Using Recursion

```
#include <stdio.h>
#include <conio.h>
int sum(int);
void main()
{
    int number, result;
    printf("Enter a positive integer: ");
    scanf("%d", &number);
    result = sum(number);
    printf("sum = %d", result);
    return 0;
}

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



Example 2: Write a recursive function to calculate the factorial value of given number.

```
#include<stdio.h>
#include<conio.h>
int fact (int);
void main()
{
    int x,y;
    clrscr();
    printf("Enter any number\n");
    scanf("%d",&x);
    y=fact(x);
    printf("Factorial value of given number====%d",y);
    getch();
}
int fact (int n)
{
    int f=1;
    if(n==1)
        return(1);
    else
        f=n*fact(n-1);
    return (f);
}
```

Example 3: Write a recursive function to check any number is Prime or not.

```
#include <stdio.h>
#include<conio.h>
int prime (int);
main()
{
    int no,i;
    printf("enter any number");
    scanf("%d",&no);
    i=prime(no);
    if(i==1)
        printf("Prime Number");
    else
        printf("Not Prime Number");
    getch();
}
int prime (int n)
{
    static int x=2;
    if(x==n)
        return(1);
    if(n%x==0)
        return(0);
    else
    {
        x++;
        prime(n);
    }
}
```

Example 4: Write a recursive function print the Fibonacci series.

```
#include<stdio.h>
#include<stdio.h>
int Fibonacci(int);
void main()
{
    int n, i = 0, c;
    scanf("%d",&n);
    printf("Fibonacci series\n");
    for ( c = 1 ; c <= n ; c++ )
    {
        printf("%d\n", Fibonacci(i));
        i++;
    }
    getch();
}
int Fibonacci(int n)
{
    if ( n == 0 )
        return 0;
    else if ( n == 1 )
        return 1;
    else
        return ( Fibonacci(n-1) + Fibonacci(n-2) );
}
```

Try Yourself:

1. What is Recursion?
2. Write a recursive program to print the table of any input number.