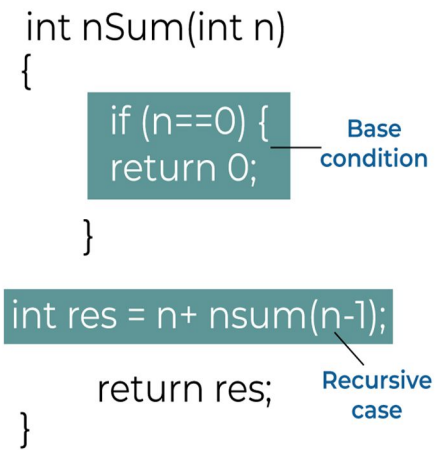
Recursion in C

Recursion is a process in programming where a function calls itself to solve smaller instances of the same problem until it reaches a base case (the stopping condition). It breaks down complex problems into smaller, more manageable sub-problems.

Key Concepts:

1. *Base Case:* Every recursive function must have a base case to prevent infinite recursion. The base case is the condition under which the recursion stops.

2. *Recursive Case:* This is the part of the function where the function calls itself with modified arguments, gradually approaching the base case.



How Recursion Works:

- The problem is divided into sub-problems.

- The function calls itself with updated arguments.

- Once the base case is reached, the function begins to return, and the results are combined as it "unwinds" back to the initial call.

Example 1: Factorial using Recursion

FACTORIAL OF N

n! = n * (n-1) * (n-2) * (n-3) * (n-4) * (n-5).....1

Example:

5! = 5*4*3*2*1 = 120

The factorial of a non-negative integer n is the product of all positive integers less than or equal to n. It is denoted by n!.

Factorial of n:

- n! = n * (n-1) * (n-2) * ... * 1

- Base Case: n = 0, 0! = 1

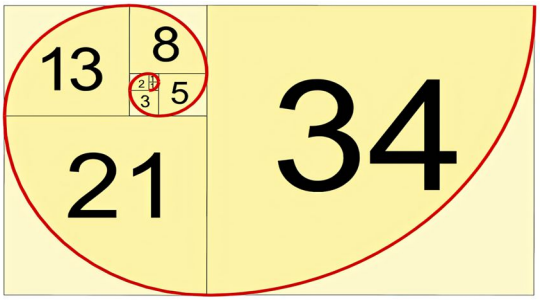
C Program to Calculate Factorial Using Recursion:

```
#include <stdio.h>

int factorial(int n) {
    if (n == 0) {
        return 1; // Base case
    } else {
        return n * factorial(n - 1); // Recursive case
    }
}

int main() {
    int num;
    printf("Enter a number: ");
    scanf("%d", &num);
    printf("Factorial of %d is %d\n", num, factorial(num));
    return 0;
}
```

Explanation:

- The factorial function calls itself with n-1 until n becomes 0 (the base case), after which it starts returning the computed values.

Example 2: Fibonacci Series using Recursion



The Fibonacci sequence is a series of numbers where each number is the sum of the two preceding ones. It starts with 0 and 1.

Fibonacci of n:

- Fib(0) = 0

- Fib(1) = 1

- Fib(n) = Fib(n-1) + Fib(n-2) for n > 1

C Program to Print Fibonacci Sequence Using Recursion:

```
#include <stdio.h>
int fibonacci(int n) {
   if (n == 0) {
      return 0; // Base case
   } else if (n == 1) {
      return 1; // Base case
   } else {
      return fibonacci(n - 1) + fibonacci(n - 2); // Recursive case
   }
}
int main() {
   int n, i;
   printf("Enter the number of terms: ");
   scanf("%d", &n);
   printf("Fibonacci Series: ");
   for (i = 0; i < n; i++) {
      printf("%d ", fibonacci(i));
   }
   printf("\n");
   return 0;
}
```

Explanation:

- The fibonacci function calls itself twice: once for n-1 and once for n-2. The function continues until it reaches the base cases n == 0 or n == 1.