C Recursion

Topics : C
Written on April 13, 2023

Recursion is a powerful technique in programming that involves a function calling itself. In C programming language, a recursive function is a function that calls itself either directly or indirectly to solve a problem. Recursive functions are used in problems where the solution depends on solutions to smaller instances of the same problem.

The general idea of recursion is to break a problem into smaller sub-problems that can be solved by calling the same function again and again until the sub-problems become simple enough to be solved directly. The recursion continues until a base case is reached, which is a problem that can be solved without further recursion.

A recursive function in C must have two parts: a base case and a recursive case. The base case specifies the problem that can be solved without further recursion, and the recursive case specifies the problem that can be broken down into smaller sub-problems.

Here is an example of a recursive function in C that calculates the factorial of a given number:

```
int factorial(int n) {
    if (n == 0) {
        return 1; // base case
    } else {
        return n * factorial(n - 1); // recursive case
    }
}
```

This function takes an integer `n` as input and calculates the factorial of `n` using recursion. The base case is when `n` is equal to 0, in which case the function returns 1. The recursive case is when `n` is greater than 0, in which case the function calls itself with `n - 1` as the argument and multiplies the result with `n`.

For example, `factorial(5)` would call itself with `factorial(4)`, which in turn would call itself with `factorial(3)`, and so on, until `factorial(0)` is reached, at which point the recursion stops and the function returns 1. The final result would be the product of all the numbers from `1` to `5`, which is `120`.