

Course
&
Test Series

Introduction to 'C' Language

Functions in C - Module - 5

Recursion in C Lang

 CBSE

Recursion in C Language

Definition: Recursion in C refers to the process where a function calls itself directly or indirectly to solve a problem. It allows a problem to be broken down into smaller sub-problems, and the function solves each sub-problem in the same manner. Recursion typically requires a **base case** to stop the function from calling itself endlessly.

 ICSE

How Recursion Works:

1. **Base Case:** This is the condition under which the recursion stops. Without a base case, recursion would continue indefinitely.
2. **Recursive Case:** This is where the function calls itself to solve a smaller version of the problem.

 NTSE

 Banking & Insurance

Example of Recursion in C:

A classic example of recursion is calculating the factorial of a number.

```
c Copy code

#include <stdio.h>

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

int main() {
    int num = 5;
    printf("Factorial of %d is %d\n", num, factorial(num));
    return 0;
}
```

Explanation:

- Factorial of n ($n!$) is calculated as $n * (n-1) * (n-2) * \dots * 1$.
- In the function `factorial()`, the base case is when `n == 0`, which returns 1 (since $0! = 1$).
- The recursive case is when the function calls itself with `n-1`, gradually reducing the problem size until it reaches the base case.

 Central Govt. Service

 State Govt. Services

 LAW Entrance

 MBA Entrance

 Railways & Metro Services

...many more

abhyasonline.in

Steps of the Function Call:

For factorial(5):

1. factorial(5) calls factorial(4)
2. factorial(4) calls factorial(3)
3. factorial(3) calls factorial(2)
4. factorial(2) calls factorial(1)
5. factorial(1) calls factorial(0) (base case)
6. factorial(0) returns 1, then the function unwinds:
 - o factorial(1) returns $1 * 1 = 1$
 - o factorial(2) returns $2 * 1 = 2$
 - o factorial(3) returns $3 * 2 = 6$
 - o factorial(4) returns $4 * 6 = 24$
 - o factorial(5) returns $5 * 24 = 120$

Thus, factorial(5) returns 120.

Advantages of Recursion:

- **Simplicity:** Recursive solutions are often easier to understand and implement for problems that have a repetitive structure (e.g., tree traversal, factorial calculation).
- **Reduction in Code Length:** Complex algorithms like tree operations or divide-and-conquer can be implemented with fewer lines of code using recursion.

Disadvantages of Recursion:

- **Memory Usage:** Every recursive function call uses stack space. Deep recursion (many function calls) can lead to a stack overflow.
- **Performance:** Recursive calls can be less efficient due to the overhead of repeated function calls, especially for problems that can be solved with iteration.

	CBSE
	ICSE
	NTSE
	Banking & Insurance
	Central Govt. Service
	State Govt. Services
	LAW Entrance
	MBA Entrance
	Railways & Metro Services
	...many more
abhyasonline.in	