

Recursion in C

Find the Sum of the First 5 Natural Numbers

Problem

Find the sum: $1 + 2 + 3 + 4 + 5 = 15$.

1. Without Recursion

A normal function can solve the problem using a loop. The function does not call itself.

```
#include <stdio.h>

int sum(int n)
{
    int i, total = 0;

    for(i = 1; i <= n; i++)
        total = total + i;

    return total;
}

int main()
{
    int result = sum(5);

    printf("Sum = %d", result);
    return 0;
}
```

Output: Sum = 15

2. With Recursion

The same problem can be solved recursively because $\text{sum}(5)$ contains smaller problems of the same form: $\text{sum}(4)$, $\text{sum}(3)$, ..., $\text{sum}(1)$.

```
#include <stdio.h>

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

int main()
{
    int result = sum(5);

    printf("Sum = %d", result);
    return 0;
}
```

Output: Sum = 15

3. What Is Recursion?

Recursion is a technique in which a function calls itself to solve a smaller or simpler version of the same problem.

Base condition

Every recursive solution needs a stopping condition. Here, $\text{if}(n == 0)$ stops further calls.

4. Going Deeper: Function Calls

Calling `sum(5)` creates a chain of recursive calls:

```
sum(5)
→ 5 + sum(4)
→ 5 + 4 + sum(3)
→ 5 + 4 + 3 + sum(2)
→ 5 + 4 + 3 + 2 + sum(1)
→ 5 + 4 + 3 + 2 + 1 + sum(0)
```

Function call stack

Each function call creates a new stack frame. The stack grows as the recursive calls go deeper.

5. Unwinding: Returning Back

When `sum(0)` returns 0, the calls return in reverse order. The stack starts shrinking.

```
sum(0) = 0
sum(1) = 1 + 0 = 1
sum(2) = 2 + 1 = 3
sum(3) = 3 + 3 = 6
sum(4) = 4 + 6 = 10
sum(5) = 5 + 10 = 15
```

LIFO — Last In, First Out

The last recursive call placed on the stack is the first one to return. This is similar to removing plates from the top of a stack.

6. Key Points to Remember

- A recursive function calls itself.
- The problem should reduce to a smaller version of itself.
- A base condition is required to stop recursion.
- Each recursive call creates a stack frame.
- The stack grows while calls go deeper.
- After the base condition, the calls return in reverse order.
- Function calls follow LIFO (Last In, First Out).

One-line idea

Going deeper → reach base condition → return back (unwind) → get the final answer.



Keep Learning • Keep Practicing • Keep Growing

EngineersTutor.com | YouTube: EngineersTutor