



# Sum of first n numbers

## Problem Statement:

Write a function `sum(n)` that calculates the sum of the first `n` natural numbers using recursion.

## Example:

**Input:** 5

**Process:**  $5 + 4 + 3 + 2 + 1 = 15$

**Output:** 15

## Concepts:

**Recursion:** A technique where a function calls itself with a reduced subproblem.

**Base Case:** Stops recursion to prevent **infinite calls**. Here, if `n === 0`, return 0.

**Recursive Case:** Return `n + sum(n - 1)`.

## Approach:

Use recursion to **reduce** the problem.

**Base case:** if `n === 0`, return 0.

**Recursive case:** return `n + sum(n-1)`.

This keeps adding numbers until `n` reaches 0, giving the total `sum`.

## Time & Space Complexity:

**Time Complexity:**  $O(n)$  one call per value from `n` to 0.

**Space Complexity:**  $O(n)$  due to call stack in recursion.

JavaScript

Python

Java

C++

C

C#

```
function sum(n) {  
  if (n === 0) return 0;  
  return n + sum(n - 1);  
}  
console.log(sum(5)); // 15
```

Video

Course

Discuss doubts

Contribute

Certificate

## Function to calculate sum of first n natural numbers using recursion - DSA Notes

Function to calculate sum of first n natural numbers using recursion - DSA Notes