Sum of all numbers in array using Recursion

Problem Statement:

Write a function **sum(n)** that calculates the sum of all numbers in an **array** arr using **recursion**. It **sums** from `index 0 to n`.

Example:

Input: [5, 2, 6, 1, 3]

Process: 5 + 2 + 6 + 1 + 3 = 17

Output: 17

Concepts:

Recursion: The function keeps **summing the element** at index `n` and `calls` itself with `n-1`.

Base Case: If `n == 0`, **return** the `first element`.

Recursive Case: `Return arr[n] + sum(n - 1).`

Approach:

If `n == 0`, **return arr[0]**.

Otherwise, return `arr[n] + sum(n - 1)`.

Time & Space Complexity:

Time Complexity: O(n) `one recursive` call per element.

Space Complexity: O(n) Due to call stack.

JavaScript Python Java C++ C C#

```
let arr = [5, 2, 6, 1, 3];

    function sum(n) {
    if (n === 0) return arr[0];
    return arr[n] + sum(n - 1);
    }
console.log(sum(arr.length - 1)); // 17
```


Sum of all numbers in array using Recursion -DSA Notes

Sum of all numbers in array using Recursion