

Sum of Array

(Recursion)

Ex : $N=4$ ✓
 $arr = \{ \underline{1}, \underline{2}, \underline{3}, \underline{4} \}$ ✓

Ans : 10

Ex : $N=3$]
 $arr = \{ \underline{4}, \underline{1}, \underline{3} \}$]

Ans : 8

✓

inp: $N=4$

$arr[] = \{1, 2, 3, 4\}$
 ↑ ↑ ↑ ↑

o/p: 10

→ Solution:

```
for (int i=0 ; i<N ; i++)  
{  
    ans += arr[i];  
}  
  
cout << ans << endl;
```

int ans = 0

Recursive Approach

I/p : $N=4$
 $arr[] = \{0, 1, 2, 3\}$
 $\{1, 2, 3, 4\}$

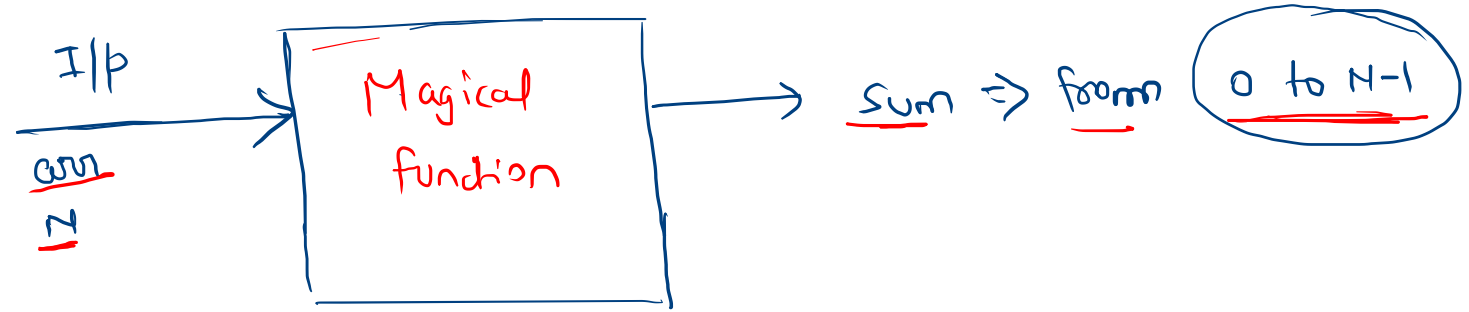
O/p : 10

\therefore Recursive function \rightarrow I/p : N , arr
 \downarrow \downarrow
integer Array

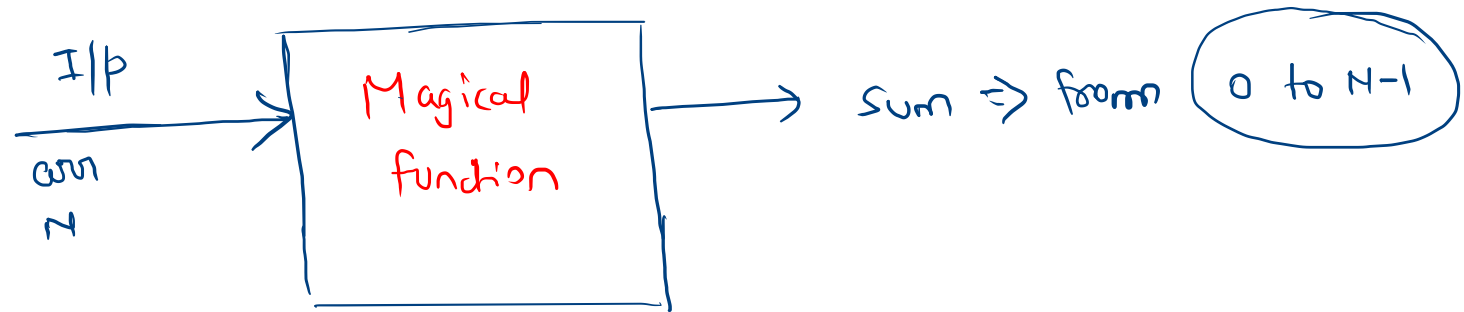
O/p : Sum of all numbers from
index $[0 \text{ to } N-1]$

\rightarrow Integer

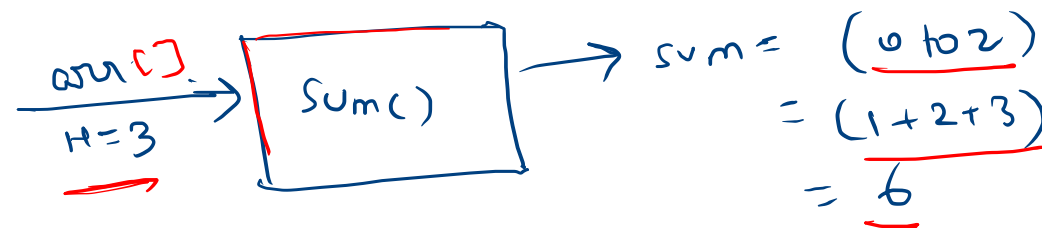
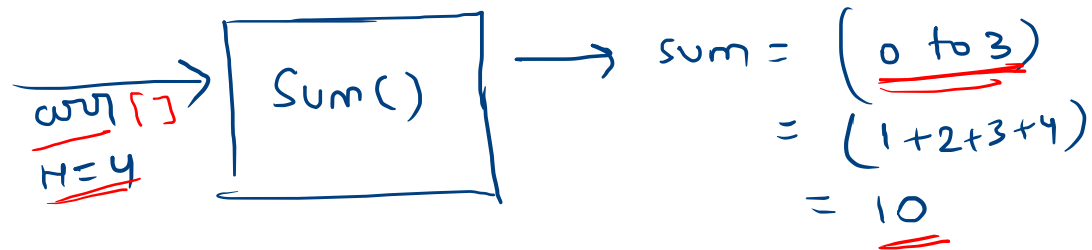
Recursive
function ✓



Recursive function



means, I/p : $N=4$
arr = $\{0, 1, 2, 3\}$



Ip : $N=4$
 $arr[] = \{1, 2, 3, 4\}$

$Sum(arr, 3) + 4 \rightarrow \underline{O/p}$

function

$Sum([1, 2, 3, 4], 4) \rightarrow \begin{matrix} 0, N-1 \\ 0, 3 \end{matrix}$

$\rightarrow \underline{O/p : 10}$

$Sum([1, 2, 3, 4], 3) \rightarrow \begin{matrix} 0, N-1 \\ 0, 2 \end{matrix}$

$\rightarrow \underline{O/p : 6}$

Ex: $N=4$

$arr[] = \{1, 2, 3, 4\}$

function

int Sum (int arr[], int N)

> // Base Condⁿ

> return arr[N-1] + sum(arr, N-1);

Go!h

Base
Condⁿ:

→ Smallest Valid I/p → check o/p

Sum(arr[], N)

{

}

→ Variable

N → 0, 1, 2, 3, ...

arr = < > N = 0

< 1 >, N = 1

< 1, 2 >, N = 2

< 1, 2, 3 >

✓ what if N = 0
→ Sum = 0

∴ No element means, Sum = 0

Recursive
solution

```
// int sum ( int arr[], int N )  
{  
    if ( N == 0 ) return 0; // Base cond?  
    return arr[N-1] + sum ( arr, N-1 )  
}
```

No. of elements
upto which we
want Total Sum

↑

//