

RECURSION IN ARRAYS

Display Array

Given: \rightarrow n : size of array a

n numbers: elements of array a

\therefore Print the elements of array from beginning to end each in separate line.

DON'T USE ITERATION AT ALL

$arr = 10, 20, 30, 40, 50$
 0 1 2 3 4 \rightarrow indexes

Expectation

displayarr(arr, 0) = 10
 20
 30
 40
 50

isse expectation hai ki yeh iss array ke 0 se 5 ki indexes ke saare elements print krdega!

Faith

displayarr(arr, 1) = 20
 30
 40
 50

isspe humne yeh faith banaliya ki yeh 1 se 5 indexes tak ke saare elements print krdega!

Expectation meets Faith

displayarr(arr, 0)

= 10 ^{arr} displayarr(arr, 1)

\therefore 10 ke khud print krdo aur displayarr(arr, 1) ko ~~chala~~ chala do!

```
public static void main (String[] args) {
```

```
    Scanner s = new Scanner (System.in);
```

```
    int n = s.nextInt();
```

```
    int [] arr = new int [n];
```

```
    for (int i = 0; i < arr.length; i++) {
```

```
        arr[i] = s.nextInt();
```

```
    }
```

```
    displayArr (arr, 0);
```

```
}
```

```

P s v displayArr (int [] arr, int idx) {
    if (idx == arr.length) { } Base Case
        return;
    }
    syso (arr[idx]); ①
    displayArr (arr, idx+1); ②
}

```

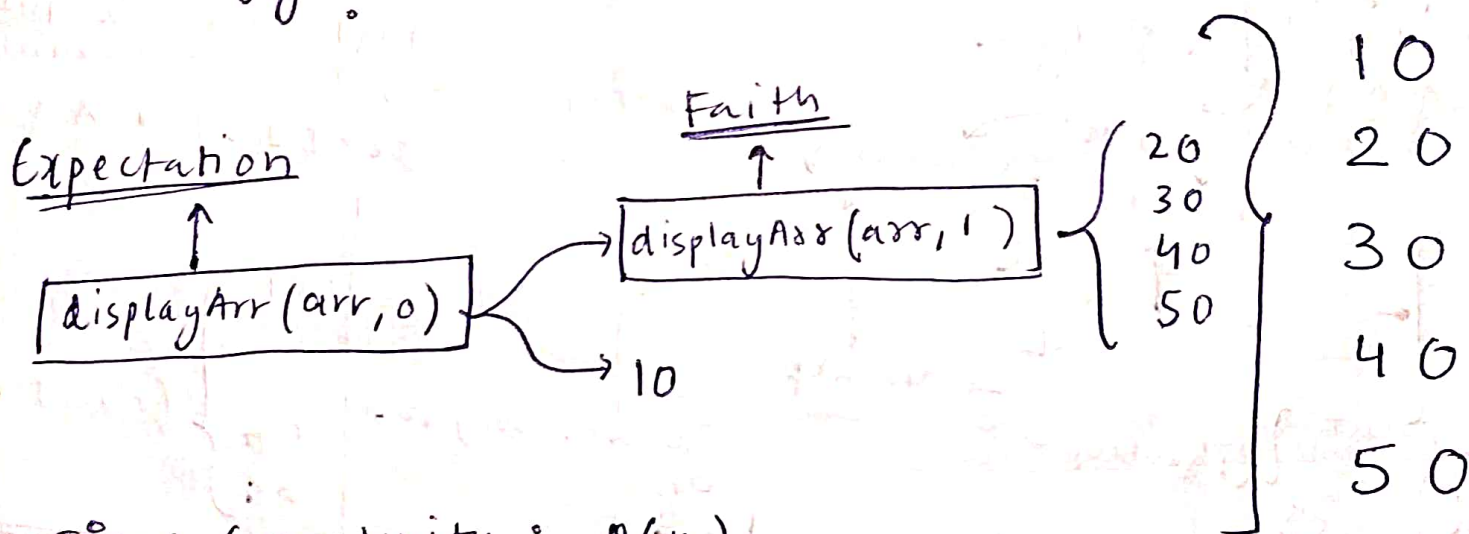
	4K	5
	4K	4
da	4K	3
da	4K	2
da	4K	1
da	4K	0
main	4K	
	arr	idx

① ② call
 ① ② call
 ① ② call
 ① ② call
 ① ② call

4K	10	20	30	40	50
	0	1	2	3	4 → indexes

4K[0] = 10 4K[3] = 40
 4K[1] = 20 4K[4] = 50
 4K[2] = 30

- Pheli line chalegi → toh arr[idx] print hoga
- Dusri line chaleny pe → call lagegi :- idx+1 hojayege
- Ans Jab base case : idx = arr.length → tab return krdeny!



Time Complexity : $O(n)$

Space Complexity : $O(n)$