

Day - 76Recursion - 78

*

Print Array

0	1	2	3	4
3	7	6	2	8

→ arr

⇒

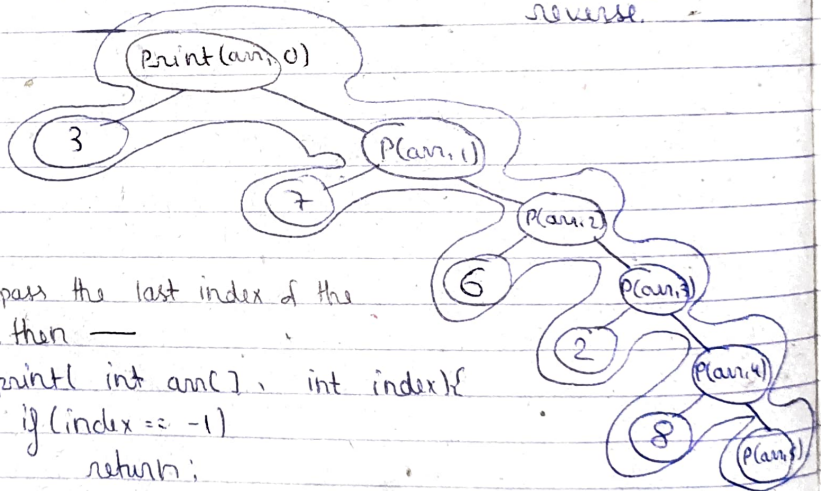
```

void print (int arr[], int index, int n) {
    if (index == n) return;
    cout << arr[index];
    print(arr, index+1);
}

```

}

Swap these two lines for printing array in reverse.



⇒

If we pass the last index of the array then —

```

void print (int arr[], int index) {
    if (index == -1) return;
    print
    cout << arr[index];
    print(arr, index-1);
}

```

}

* Sum of all element

0	1	2	3	4
3	4	5	8	2

→ arr

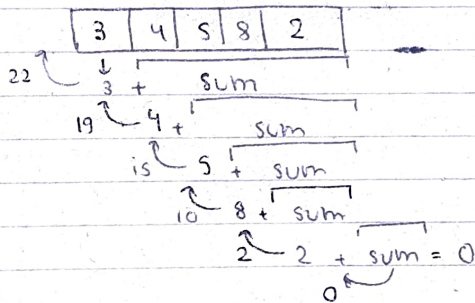
⇒

```

int sum (int arr[], int index, int n){
    if (index == 0)
        return 0;
    return arr[index] + sum(arr, index+1, n);
}

```

⇒ Here, we are first finding the sum of index 0 then the sum of other elements.



⇒

```

sum(1) = 0
sum(4) = 2 + sum(1)
sum(3) = 8 + sum(4)
    |
sum(index) = arr[index] + sum(index-1);

```

* Min element

0	1	2	3	4
7	2	4	1	6

=> Here,

=> We will do —

$$\min(7, 2) = 2$$

$$\min(2, 4) \rightarrow 2$$

$$\min(2, 1) \rightarrow 1$$

$$\min(1, 6) \rightarrow 1$$

int n

```

=> int min(int arr, int index){
    if(index == arr.size() n - 1)
        return arr[index];
    return min(arr[index], min(arr, index+1));
}

```