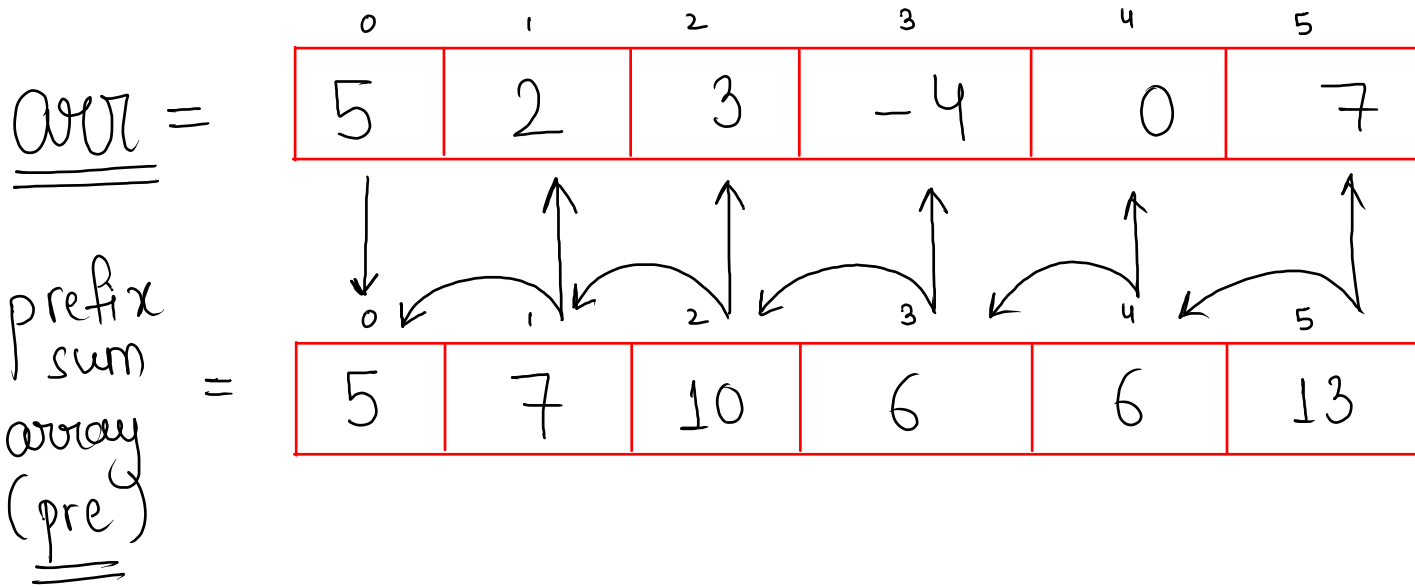


$\Rightarrow$  Prefix Sum (sum of all the elements on the left side including itself)



eg<sup>n</sup>

$$\text{pre}[i] = \text{arr}[i] + \text{pre}[i-1]$$

code

0	1	2	3	4	5
5	2	3	-4	0	7

arr =

pre[0] = arr[0];

for (int i = 1; i < n; i++) {

pre[i] = arr[i] + pre[i-1]

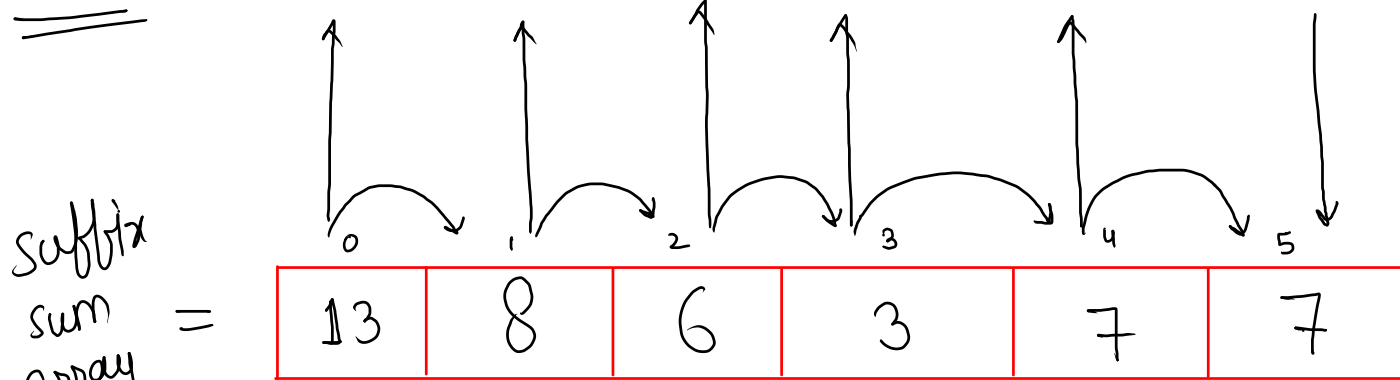
pre =

0	1	2	3	4	5
5	7	10	6	6	13
	2+5	3+7	-4+10	0+6	7+6

$\Rightarrow$  Suffix Sum (sum of all the elements on right side including itself)

arr =

0	1	2	3	4	5
5	2	3	-4	0	7



(suffix)

eg.)

$$\text{suffix}[i] = \text{arr}[i] + \text{suffix}[i+1];$$

code

n=6

suffix[n-1] = arr[n-1];

for(int i = n-2 ; i >= 0 ; i--) {

suffix[i] = arr[i] + suffix[i+1];

}

code

$$T.C = \underline{\underline{O(n)}}$$

```
public class Main {  
    public static void main(String[] args) {  
        int[] arr = {5, 2, 3, -4, 0, 7};  
        int n = arr.length;  
        int[] pre = new int[n];  
        pre[0] = arr[0];  
        for (int i = 1; i < n; i++) {  
            pre[i] = arr[i] + pre[i - 1];  
        }  
  
        for (int i = 0; i < n; i++) {  
            System.out.print(pre[i] + " ");  
        }  
    }  
}
```

# Greatest Till Me

arr =

0	1	2	3	4	5
5	2	3	-4	0	7

prefix  
array  
(pre)

0	1	2	3	4	5
5	5	5	5	5	7

eg.)

$$pre[i] = \text{Math.max}(arr[i], pre[i-1]);$$

code

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];
    for (int i = 0; i < n; i++) {
        arr[i] = scn.nextInt();
    }

    greaterTillMe(arr, n);
}

public static void greaterTillMe(int[] arr, int n) {
    int[] pre = new int[n];
    pre[0] = arr[0];
    for (int i = 1; i < n; i++) {
        pre[i] = Math.max( arr[i], pre[i - 1] );
    }

    // print
    for (int i = 0; i < n; i++) {
        System.out.println(pre[i]);
    }
}
```

T.C =  $O(n)$

S.C =  $O(n)$

# Print Prefix Sum between L and R

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];
    for (int i = 0; i < n; i++) {
        arr[i] = scn.nextInt();
    }
    int left = scn.nextInt();
    int right = scn.nextInt();
    printPrefix(arr, n, left, right);
}

public static void printPrefix(int[] arr, int n, int left, int right) {
    int[] pre = new int[n];
    pre[0] = arr[0];
    for (int i = 1; i < n; i++) {
        pre[i] = arr[i] + pre[i - 1];
    }

    for (int i = left; i <= right; i++) {
        System.out.println(pre[i]);
    }
}
```

$T.C = O(n)$

$S.C = O(n)$



# Find Pivot Index 1

(gmp)

arr =

0	1	2	3	4	5	6	7
5	-3	2	7	1	1	1	1
↓	↓	↓	↓				
$l = 0$	$l = 5$	$l = 2$	$l = 4$				
$r = 10$	$r = 13$	$r = 11$	$r = 4$				

ans = 3

arr =

0	1	2	3	4	5	6	7
5	-3	2	7	1	1	1	1

pre =

0	1	2	3	4	5	6	7
5	2	4	11	12	13	14	15

suffix =

0	1	2	3	4	5	6	7
15	10	13	11	4	3	2	1

check if  $pre[i] == suffix[i]$

Note:- if prefix of  $i$  is equal to  
suffix of  $i$  then  $i$  will be  
our answer.

Code

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];
    for (int i = 0; i < n; i++) {
        arr[i] = scn.nextInt();
    }

    System.out.println(findPivot(arr, n));
}

public static int findPivot(int[] arr, int n) {
    int[] pre = new int[n];
    pre[0] = arr[0];
    for (int i = 1; i < n; i++) {
        pre[i] = arr[i] + pre[i - 1];
    }

    int[] suffix = new int[n];
    suffix[n - 1] = arr[n - 1];
    for (int i = n - 2; i >= 0; i--) {
        suffix[i] = arr[i] + suffix[i + 1];
    }

    for (int i = 0; i < n; i++) {
        if (pre[i] == suffix[i]) {
            return i;
        }
    }
    return -1;
}
```

$$T.C = O(n)$$

$$S.C = O(n)$$