Q1 Given n array elements and a starting index and a ending index. Find sum in that range.

Ex1 : arr[] =
$$\begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \\ \end{array}$$
$$[\,2 \quad 4 \quad 1 \quad 6 \quad 5\,]$$

$l = 2$

$r = 4$

⇒ loop from $l - r$, find sum

TC: $O(n)$

SC: $O(1)$

Pseudo Code :

```
int sum = 0
for (int i = l ; i <= r ; i++) {
    sum += arr[i];
}
return sum;
```

Q2 Given n array elements and a starting index and a ending index. Find sum in that range. You are given Q queries.

$$
\begin{array}{ccccccc}
 & & 0 & 1 & 2 & 3 & 4 \\
arr[\,] & = & [\,2 & 4 & 1 & 6 & 5\,]
\end{array}
$$

$Q = 3$

1) $l = 0, \ r = 3 \Rightarrow 13$

2) $l = 1, \ r = 2 \Rightarrow 5$

3) $l = 2, \ r = 4 \Rightarrow 12$

```
for (int j = 0 ; j < q ; j++) {
        // l, r as input;
        int sum = 0
        for ( int i = l ; i <= r ; i++) {
                Sum + = arr [i];
        }
        print (sum);
}
```

TC : O(QN)

Q3 Given are Indian score after every over.

Overs	1	2	3	4	5	6	7	8	9	10
Runs S[ ]	2	8	14	29	31	49	65	79	88	97

1) Runs scored in 10^{th} over : [10 , 10]

$S[10] - S[9]$

$97 - 88 = 9$

2) Runs scored in the last 5 overs. : [6, 10]

$S[10] - S[5]$

$97 - 31 = 66$

3) Runs scored in 7^{th} over : [7, 7]

$S[7] - S[6]$

$65 - 49 = 16$

Cumulative sum from the start $\Rightarrow$ Prefix Sum

Prefix Sum Array.

pf[i] $\Rightarrow$ Sum of all elements from the start till i^{th} index.

$$\sum_{x=0}^{i} arr[i]$$

arr[ ] =

0	1	2	3	4	5
-9	10	2	1	8	5

pf[ ] =

0	1	2	3	4	5
-9	1	3	4	12	17

1) Sum(2,4) =

Sum(0,4) = Sum(0,1) + Sum(2,4)

Sum(2,4) $\Rightarrow$ Sum(0,4) − Sum(0,1)

Sum(2,4) $\Rightarrow$ pf[4] − pf[1]

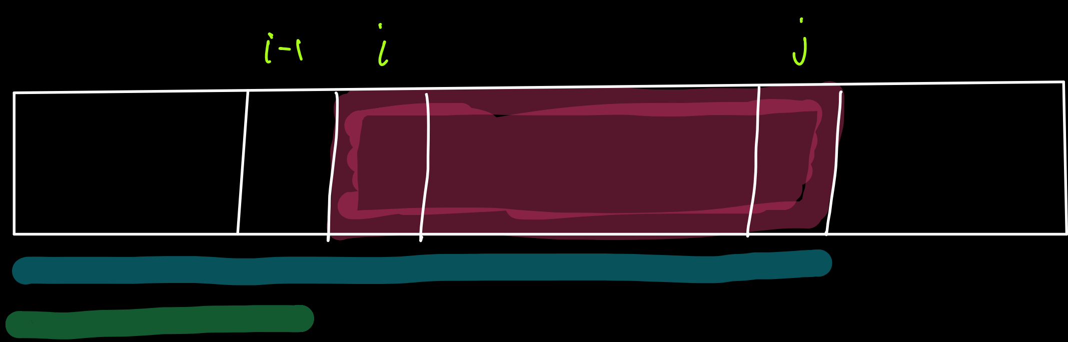
12 − 1 $\Rightarrow$ 11

2) Sum $(1, 3)$ = pf $[3]$ - pf $[0]$

3) Sum $(2, 3)$ = pf $[3]$ - pf $[1]$

Sum (i, j)

pf[]:



$$\boxed{\text{Sum}(i, j) = pf[j] - pf[i-1]}$$

$(0, 3)$ = pf $[3]$ [Edge Case]

```
# if (i==0)
      Sum = pf[j]
  else
      Sum = pf[j] - pf[i-1]
```

Revisit

Given n array elements and a starting index and a ending index. Find sum in that range. You are given Q queries.

```
// prepare prefix sum array.
pf[0] = arr[0];
for (int i=1; i<n; i++) {
    pf[i] = pf[i-1] + arr[i];
}
for (int i=0; i<q; i++) {
    // l, r are input.
    int sum;
    if (l == 0)
        sum = pf[r];
    else
        sum = pf[r] - pf[l-1];
    print sum;
}
```

$O(n)$

$O(q)$

n

q

nq

n+q

1

TC: $O(n+q)$

SC: $O(n)$

1) Can we think in terms of $n > q$ or $q > n$

$$TC: O(n+q)$$

2) Can we modify the same array

→ depends if the original array is later needed

```
for (int i = 1; i < n; i++) {
    arr[i] = arr[i-1] + arr[i];
}
```

$SC: O(1)$ → if we modify original

$: O(n)$ → if we don't

Equilibrium Index

$\Rightarrow n \geq 1$

Given an arr[], find count of Equilibrium Index.

Equilibrium Index $\Rightarrow$ Sum of all elements to left is equal to Sum of all elements to right

	0	1	2	3
arr[] =	-3	2	4	-1
l =	0	-3	-1	3
r =	5	3	-1	0

ans = 1

left $\Rightarrow$ pf[i-1]

right $\Rightarrow$ [i+1, n-1]

pf[n-1] - pf[i]

$i = 0 \Rightarrow$ Edge left $= pf[-1]$

left $\Rightarrow 0$

$i = n-1 \Rightarrow$ right $= 0$

$pf[n-1] - pf[n-1] = 0$

$n = 1 \Rightarrow pf[0] - pf[0] = 0$

```
int cnt=0;  int left, right;
for (int i=0 ; i<n ; i++) {

    if (i==0)
        left=0
    else
        left = pf[i-1]

    right = pf[n-1] - pf[i]
    if (left==right)
        cnt++;
}

return cnt;
```

TC: $O(n)$

SC → $O(1)$ / $O(n)$

If we modify original array.

Q4 Given N array elements & Q queries.

Each Q queries consists of i, j.

Return the number of even valued terms in [i, j]

Ex1 arr[] =

$$\begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \\ \end{array}$$
$$[\,2 \quad 3 \quad 6 \quad 8 \quad 1 \quad 10\,]$$

(i) [0, 3] ⇒ 3

(ii) [4, 5] ⇒ 1 !

$$\begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 6 & 8 & 1 & 10 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 2 & 3 & 3 & 4 \end{array}$$

[0, 3] ⇒ pf[3] ⇒ 3

[4, 5] ⇒ pf[5] − pf[3] ⇒ 1

1) Update your array.

even element = 1 $O(n)$

odd element $\Rightarrow 0$

2) Construct pf on the updated array $\Rightarrow O(n)$.

3) Solve for each query $\Rightarrow O(q)$

TC: $O(n+q)$

SC

$O(1)$ if we modify original array

$O(n)$

leetcode → prefix Sum