Q → Given an integer array, find sum of elements from index L to index R.

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

L = 1   R = 3

Ans = 6 + 2 + 4 = 12

```
sum = 0
for i → L to R {
    sum += A[i]
}
return sum
```

TC = O(N)   SC = O(1)

Solve the above for multiple queries → (L, R)

same task for multiple inputs

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

Queries → (1, 3) , (2, 2) , (1, 5)

Ans → 12, 2, 19

$$\begin{array}{ccccc} & 0 & 1 & 2 & \\ L = [& 1 & 2 & 1 &] \\ R = [& 3 & 2 & 5 &] \end{array}$$

```
for i → 0 to (Q-1) {
    sum = 0
    for j → L[i] to R[i] {
        sum += A[j]
    }
    print(sum)
}
```

TC = O(Q * N)   SC = O(1)

Scoreboard in cricket

Over →	0	1	2	3	4	5	6	7	8	9	10
Score →	0	2	8	14	29	31	49	65	79	88	97

Runs in 7^{th} over → $65 - 49 = \underline{16}$

Runs from 6^{th} to 10^{th} over → $97 - 31 = \underline{66}$

Runs in 10^{th} over → $97 - 88 = \underline{9}$

Runs from 3^{rd} to 6^{th} over → $49 - 8 = \underline{41}$

Runs from 4^{th} to 9^{th} over → $88 - 14 = \underline{74}$

★ sum from index L to R → (sum from 0 to R) − (sum from 0 to (L−1))

prefix sum

prefix sum

$$P[i] = \sum_{u=0}^{i} A[u] \qquad A[0] + A[1] + \ldots A[i]$$

sum from index L to R = $P[R] - P[L-1]$

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

$P[2] = -3 + 6 + 2$

$P[3] = \underbrace{-3 + 6 + 2}_{P[2]} + 4$

$P[3] = P[2] + A[3]$

$$\boxed{\begin{array}{l} P[i] = P[i-1] + A[i] \\ P[0] = A[0] \end{array}}$$

$$\begin{array}{ccccccc} & 0 & 1 & 2 & 2 & 3 & 4 \\ A \rightarrow & [10 & 32 & 6 & 12 & 20 & 1] \\ P \rightarrow & 10 & 42 & 48 & 60 & 80 & 81 \end{array}$$

P[0] = A[0]

for i ⟶ 1 to (N-1) {
    P[i] = P[i-1] + A[i]
}

for i ⟶ 0 to (Q-1) {
    l = L[i]   r = R[i]
    if (l > 0)
        sum = P[r] - P[l-1]
    else sum = P[r]
    print(sum)
}

TC = $\underline{O(N+Q)}$ SC = $\underline{O(N)}$ ⟶ $\underline{P[N]}$

Can it be optimized?

$$
\begin{array}{lcccccc}
 & 0 & 1 & 2 & 2 & 3 & 4 \\
A \to & [10 & \cancel{32} & \cancel{6} & \cancel{12} & \cancel{20} & \cancel{1}] \\
 & & 42 & 48 & 60 & 80 & 81
\end{array}
$$

$$\boxed{\begin{array}{l} A[i] = A[i-1] + A[i] \\ \cancel{A[0] = A[0]} \end{array}}$$

P ⟶ 10 42 48 60 80 81

for i ⟶ 1 to (N-1) {
    A[i] = A[i-1] + A[i]
}

SC = $\underline{O(1)}$

Q ⟶ Given an integer array of size N & Q queries. Find the sum of all even index elements from L to R.

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

Queries

1 3 → 1

2 5 → 5

0 4 → 7

3 3 → 0

Bruteforce → TC = O(N * Q)

SC = O(1)

Prefix sum only for even index values

$$\boxed{\begin{array}{ll} P[0] = A[0] & \\ \text{if } (i \% 2 == 0) & P[i] = P[i-1] + A[i] \\ \text{else} & P[i] = P[i-1] \end{array}}$$

$$\begin{array}{ccccccc} & 0 & 1 & 2 & 3 & 4 & \overset{i}{5} \\ A = [& 2 & 3 & 1 & 6 & 4 & 5\ ] \\ P \rightarrow & 2 & 2 & 3 & 3 & 7 & 7 \end{array}$$

$l = 2$ $r = 5$ sum = $\underline{P[5] - P[1]}$

$$\begin{array}{cccccc} & 0 & 1 & 2 & 3 & 4 \\ A = [& 2 & 4 & 3 & 1 & 5\ ] \\ P = & 2 & 2 & 5 & 5 & 10 \end{array}$$

```
P[0] = A[0]
for i → 1 to (N-1) {
    if (i % 2 == 0)    P[i] = P[i-1] + A[i]
    else       P[i] = P[i-1]
}
```

```
for i → 0 to (Q-1) {
    l = L[i]    r = R[i]
    if (l > 0)
        sum = P[r] - P[l-1]
    else  sum = P[r]
    print(sum)
}
```

$TC = \underline{O(N+Q)}$

$SC = \underline{O(N)}$

$$A = \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \\ [\,2 & 4 & 3 & 1 & 5\,] \end{array}$$

↳ 2 2 5 5 10

```
for i → 1 to (N-1) {
    if (i % 2 == 0)   A[i] = A[i-1] + A[i]
    else              A[i] = A[i-1]
}
```

$SC = \underline{O(1)}$

$$A = \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \\ [\,2 & \cancel{4} & \cancel{3} & \cancel{1} & \cancel{5}\,] \\ & 2 & 5 & 5 & 10 \end{array}$$

Prefix sum for odd index →

$$\boxed{\begin{array}{ll} P[0] = 0 & \\ \text{if } (i \% 2 == 1) & P[i] = P[i-1] + A[i] \\ \text{else} & P[i] = P[i-1] \end{array}}$$

Q → Given an integer array, count the number of special index i.e. those index after removing which, sum of all even index elements = sum of all odd index elements

i	0	1	2	3	4	5	S_e		S_o
A =	[4	3	2	7	6	-2]			
0 ✓	3	2	7	6	-2		8	=	8
1 ✗	4	2	7	6	-2		9	≠	8
2 ✓	4	3	7	6	-2		9	=	9
3 ✗	4	3	2	6	-2		4	≠	9
4 ✗	4	3	2	7	-2		4	≠	10
5 ✗	4	3	2	7	6		12	≠	10

Ans = 2

odd ⇒ even
even ⇒ odd

$$A = [\overset{0}{4}\quad \overset{1}{1}\quad \cancel{\overset{2}{3}}\quad \boxed{\overset{3}{7}\quad \overset{4}{10}}]$$

$$4\quad 1\quad \boxed{7\quad 10}$$

$S_o = 1 + 10 = \underline{11}$

$$A = [\overset{0}{2}\quad \overset{1}{3}\quad \overset{2}{1}\quad \cancel{\overset{3}{4}}^{\,i}\quad \boxed{\overset{4}{0}\quad \overset{5}{-1}\quad \overset{6}{2}\quad \overset{7}{-2}\quad \overset{8}{10}\quad \overset{9}{8}}]$$

$$2\quad 3\quad 1\quad \boxed{0\quad -1\quad 2\quad -2\quad 10\quad 8}$$

$S_o = 3 + 0 + 2 + 10$
$= \underline{15}$

$S_e = 2 + 1 + (-1) + (-2) + 8 = \underline{8}$

Check i^{th} index ⤵

$\underline{S_o}$		$\underline{S_e}$
sumOdd(0 — (i-1)) + sumEven((i+1) — (N-1))	=	sumEven(0 — (i-1)) + sumOdd((i+1) — (N-1))

oddP[i-1] + (evenP[N-1] − evenP[i])   evenP[i-1] +

(oddP[N-1] - oddP[i])

// oddP[]   evenP[] ✓

```
cnt = 0
if ((evenP[N-1] - evenP[0]) == (oddP[N-1] - oddP[0]))
    cnt++

for i → 1 to (N-1) {
    so = oddP[i-1] + evenP[N-1] - evenP[i]
    se = evenP[i-1] + oddP[N-1] - oddP[i]
    if (so == se) cnt++
}
return cnt
```

TC = O(N)   SC = O(N)

O(N + N) → O(N)

prefix L → R

suffix L ← R