$A = [\underbrace{1, 2, 3}_{K}, 4, \underbrace{5, 6, 7}_{K}]$ $K = 3$

R → L $[\underbrace{5, 6, 7}_{K}, \underbrace{1, 2, 3, 4}_{N-K}]$

L - R $[\underbrace{4, 5, 6, 7}_{N-K}, \underbrace{1, 2, 3}_{K}]$

Sol^n

1) Reverse the entire array

$[\underbrace{7, 6, 5, 4}_{N-K}, \underbrace{3, 2, 1}_{K}]$

2) Reverse ele from 0 to N-K-1

3) Reverse from N-K to N-1.

Sum of Even Indexed Elements

Q Given an array of size N & Q queries with start (s) & end (e) indexes of a range.
[e >= s]

For each query, return the sum of all even indexed elements in that range.

Eg $A = \{ \overset{0}{2}, \overset{1}{3}, \overset{2}{1}, \overset{3}{6}, \overset{4}{4}, \overset{5}{5} \}$

Query :	S,	e	Ans
1)	1	3	1
2)	2	5	A[2] + A[4] = 5
3)	3	3	0

Solⁿ 1) Brute force Approach

⇒ For every query, iterate over the range from s to e & add the even indexed elements

if (index % 2 == 0)

Code

O/P : Array of size Q

```
function evenSum (int A[], int query[][], N, Q) {
    int ans[Q];
    for (i = 0; i < Q; i++) {          // Q
        s = query[i][0];
        e = query[i][1];
        sum = 0;
```

```
for (j = s; j <= e; j++) {   // N
    if (j % 2 == 0) {
        sum = sum + A[j];
    }
}
ans[i] = sum;
}
return ans;
}
```

$T.C. = O(N \times Q)$

$S.C. = O(1)$

Const: $1 \leq N \leq 10^5$
$1 \leq Q \leq 10^5$

2) Optimise

$PS[i] \Rightarrow$ Sum of all ele. from 0 to i

$PSE[i] \Rightarrow$ Sum of all even indexed elements from 0 to i.

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

$$\begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array}$$
$$PSE = [\,2,\ 2,\ 3,\ 3,\ 7,\ 7\,]$$

$PSE[0] = \text{Sum Even}[0,0] = A[0]$

$PSE[1] = \text{Sum Even}[0,1] = A[0]$

$PSE[2] = \text{Sum Even}[0,2] = A[0] + A[2]$

$PSE[3] = \text{Sum Even}[0,3] = A[0] + A[2]$

$PSE[4] = \text{Sum Even}[0,4] = \underbrace{A[0] + A[2]}_{\Downarrow} + A[4]$

$PSE[3] + A[4]$

$PS[5] = \text{Sum Even}[0,5] = \underline{A[0] + A[2] + A[4]}$

$PSE[4]$

$$PSE[i] \xrightarrow[\text{Even}]{\text{if } (i \,\%\, 2 == 0)} PSE[i-1] + A[i]$$

$$\xrightarrow[\text{ODD}]{\text{else}} PSE[i-1];$$

```
PSE[0] = A[0];

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

Code

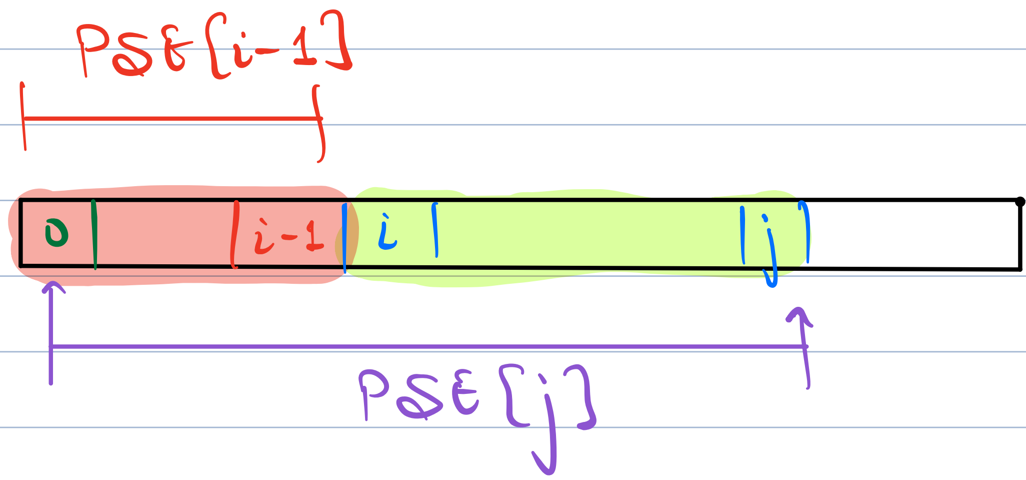
```
function evenSum(int A[], int query[][], N, Q) {

  PSE[N];
  PSE[0] = A[0];

  for (i= 1; i < N; i++) {
          if (i % 2 == 0) {
              PSE[i] = PSE[i-1] + A[i];
          } else {
              PSE[i] = PSE[i-1];
          }
  }

  int ans[Q];
```

$O(N)$

⇒ Output

```
for (i = 0; i < Q; i++) {                  // Q      O(Q)
    s = query[i][0];
    e = query[i][1];
    sum = 0;
    if (s == 0) {
        sum = PSE[e];
    } else {                               // O(1)
        sum = PSE[e] - PSE[s-1];
    }
    ans[i] = sum;
}
return ans;
}
```

T.C. = $O(N+Q)$
S.C. = $O(N)$



Ex / Quiz

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

PSE = [2, 2, 5, 5, 10]

Sum[0, 3] = PSE[3]

Sum of all Odd Indexed Ele.

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

PSO = [0, 4, 4, 5, 5]

```
PSO[0] = 0;

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

Special Index

Q) Given an integer array of size N.
Count & return the number of **Special Index** present in array.

Special Index : Indices after removing which, sum of all ODD indexed elements becomes equal to sum of all EVEN indexed elements of the array.

Eg

$$A[\,] = \{ \overset{0}{4}, \overset{1}{3}, \overset{2}{2}, \overset{3}{7}, \overset{4}{6}, \overset{5}{-2} \}$$

A[ ]

i	0	1	2	3	4	Sodd	Seven
0	3,	2,	7,	6,	-2	8	8
1	4	2	7	6	-2	8	9
2	4	3	7	6	-2	9	9
3	4	3	2	6	-2	9	4
4	4	3	2	7	-2	10	4
5	4	3	2	7	6	10	12

$Ans = 2$

Q

$$A = \begin{array}{c} \begin{matrix} 0 & 1 & 2 & 3 & 4 \end{matrix} \\ \left[\begin{matrix} 4, & 1, & \cancel{3}, & 7, & 10 \end{matrix} \right] \end{array}$$

$$\Downarrow \qquad i = 2$$

$$\begin{array}{c} \begin{matrix} 0 & 1 & 2 & 3 \end{matrix} \\ \left[\begin{matrix} 4, & \underline{1}, & 7, & \underline{10} \end{matrix} \right] \end{array}$$

$Ans = 11$

Q

$$\begin{matrix} 0 & & i-1 & i & i+1 & j & & & & N-1 \end{matrix}$$

$$A = \begin{array}{c} \begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{matrix} \\ \left[\begin{matrix} 2, & 3, & 1, & \underline{\underline{4}}, & 0, & -1, & 2, & -2, & 10, & 8 \end{matrix} \right] \end{array}$$

$$\times$$

$$A_{new} = \begin{array}{c} \begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{matrix} \\ \left[\begin{matrix} 2, & 3, & 1, & 0, & -1, & 2, & -2, & 10, & 8 \end{matrix} \right] \end{array}$$

$$\underset{A_{New}}{\text{Sum Odd}} = \underset{A}{\text{Sum Odd}}\,[0, 2] + \underset{A}{\text{Sum Even}}\,[4, 9]$$

$$3 \quad + \quad 12$$

$$= 15 \quad \underline{\underline{Ans}}.$$

$\Rightarrow$ for index $j > i \Rightarrow$ the index in the final array reduces by 1.

(Even indexed element in A) $\Rightarrow$ (Odd indexed element in A_{new})

(Odd indexed element in A) $\Rightarrow$ (Even indexed element in A_{new})

$$\text{Sum Even}_{A_{New}} = \text{Sum Even}_{A}[0, i-1] + \text{Sum Odd}_{A}[i+1, N-1]$$

$$\text{Sum Odd}_{A_{new}} = \text{Sum Odd}_{A}[0, i-1] + \text{Sum Even}_{A}[i+1, N-1]$$

Q

$$A = \begin{array}{cccccccccc} 0 & & i-1 & i & i+1 & j & & & & N-1 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \end{array}$$

$$A = [2, 3, 1, \cancel{4}, 0, -1, 2, -2, 10, 8]$$

$$\text{Sum Even}_{A_{New}} = \text{Sum Even}_{A}[0, 2] + \text{Sum Odd}_{A}[4, 9]$$

$$3 + 5$$

$$= 8 \text{ Ans.}$$

Sol^n 1) Brute force

For every element of the array (N)

$O(N) \Rightarrow$ Copy all elements except the element to be deleted in a new array of size N-1

$O(N) \Rightarrow$ Iterate & calculate sum of odd elements

$O(N) \Rightarrow$ Iterate & calculate sum of even elements

$\Rightarrow$ Check if both sums are same

$$T.C. = O\left(N \times (N+N+N)\right) = O(N^2)$$

2) Optimise

$\Rightarrow$ We do not need to copy rem. elements to a new array.

We can directly calculate the sum of Odd & Even indexed element after deletion.

⇒ for index $j > i$ ⇒ the index in the final array reduces by 1.

$\left(\begin{array}{c}\text{Even indexed element} \\ \text{in } A\end{array}\right) \Rightarrow \left(\begin{array}{c}\text{Odd indexed element} \\ \text{in } A_{new}\end{array}\right)$

$\left(\begin{array}{c}\text{Odd indexed element} \\ \text{in } A\end{array}\right) \Rightarrow \left(\begin{array}{c}\text{Even indexed element} \\ \text{in } A_{new}\end{array}\right)$

$$\underset{A_{new}}{\text{Sum Even}} = \underset{A}{\text{Sum Even}}[0, i-1] + \underset{A}{\text{Sum Odd}}[i+1, N-1]$$

$$\underset{A_{new}}{\text{Sum Odd}} = \underset{A}{\text{Sum Odd}}[0, i-1] + \underset{A}{\text{Sum Even}}[i+1, N-1]$$

⇒ Range sum of odd & even indexed elements can be calculated efficiently using Prefix Sum Technique.

⇒ Calculate PSO & PSE arrays.

$$\text{Sum Even}_{A_{New}} = \text{Sum Even}_A[0, i-1] = PSE[i-1]$$

if (i == 0) {

$= 0$

}

$$+ \qquad\qquad +$$

$$\text{Sum Odd}_A[\overset{s}{i+1}, \overset{e}{N-1}] = PSO[\overset{e}{N-1}] - PSO[\overset{s-1}{i}]$$

$$\text{Sum Odd}_{A_{new}} = \text{Sum Odd}_A[0, i-1] = PSO[i-1]$$

if (i == 0) {

$= 0$

}

$$+ \qquad\qquad +$$

$$\text{Sum Even}_A[\overset{s}{i+1}, \overset{e}{N-1}] = PSE[\overset{e}{N-1}] - PSE[\overset{s-1}{i}]$$

Code

```
function countSpecial(int A[], int N) {
    // Calculate PSO    O(N)
    // Calculate PSE    O(N)
```

```
    count = 0;
i=0 { if ((PSO[N-1] - PSO[0]) == (PSE[N-1] - PSE[0])) {
          count++;
      }

    for (i = 1; i < N; i++) {

        sumOdd = PSO[i-1] + (PSE[N-1] - PSE[i]);

        sumEven = PSE[i-1] + (PSO[N-1] - PSO[i]);

        if (sumOdd == sumEven) {
              count++;
        }
    }
    return count;
}
```

$T.C. = O(N)$

$S.C. = O(N)$