RQ-1

```
int  arr[10]      // 40
int  x            // 4
int  y            // 4
long z            // 8
int  arr[N]       // 4N
```

SC: O(N)

RQ-2.

Time for accessing an array index TC: O(1)

arr[i]

RQ-3

```
i=0, j=N-1
while(i<j)
{
    swap(arr[i], arr[j])
    i++
    j--
```

RQ-4

function (.....)
{
rev (arr, 0, n-1)
rev (arr, 0, k-1)
rev (arr, k, n-1)

k = 2

1 2 3 4 5

↓

5 4 3 2 1

↓

4 5 3 2 1

↓

4 5 1 2 3

rotating array twice

RQ-5

Fast Access element	Insert element
Array ✓	Array ✗
Array list ✓	Arraylist ✓

Q. Given stock's profit/loss for each day.

Develop a feature where you will be given start day & end day, need to return total profit/loss for this duration
↓
(Both inclusive)

	0	1	2	3	4	5	6	7	8	9
Stock[] =	[-5,	10,	20,	40,	50,	-10,	80,	-90,	-20,	-10]

Start day	End date	Net Profit/loss
1	4	120
0	0	-5
7	9	-120
0	9	65
2	7	90

Q Given N elements, Q queries. For each query,

Cal sum of elements from L to R (both inclusive)

arr =	[-3,	6,	2,	4,	5,	2,	8,	-9,	3,	1]
	0	1	2	3	4	5	6	7	8	9

L	R	Solution
4	8	9

BF: for each query L, R.

Iterate L to R, and get the sum.

↓

Not sufficient

$1 \le Q \le 10^5$

$1 \le N \le 10^5$

```
for (each query)
{
    l, r

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

TC: $O(QN)$

SC: $O(1)$

{ $\overset{4}{\_\_}$, $\overset{14}{\_\_}$, $\overset{14}{\_\_}$, $\overset{15}{\_\_}$, $\underbrace{\_\_ , . \_\_ .}$ }

↓

Team score After 1st over

Quiz 1

$\left[\ \overset{1}{2},\ \overset{2}{8},\ \overset{3}{14},\ \overset{4}{29},\ \overset{5}{31},\ \overset{6}{49},\ \overset{7}{65},\ \overset{8}{79},\ \overset{9}{88},\ \overset{10}{97}\ \right]$

Total runs after 7th over = 65

Total runs after 6th over = 49

runs in 7th over = score[7] - score[6]

= 65 - 49

= 16

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

runs in 6th to 10th = Score [10] - Score [6]

= 97 - 31

= 66

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

runs in 10th = score [10] - score [9]

= 97 - 88

= 9

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

3rd to 6th over = Score [6] - Score [2]

= 49 - 8

= 41

1 2 3 4 5 6 7 8 9 10

[2, 8, 14, 29, 31, 49, 65, 79, 88, 97]

prefix sum array

4th to 9th = score[9] - score[3]

= 88 - 14

= 74

0 1 2 3 4

Arr [ ] = [2, 5, -1, 7, 1] ← Profit/loss for each day

PF [ ] = [2, 7, 6, 13, 14]

PF[i] = sum of all elements from 0 to i

arr[] = 10, 32, 6, 12, 20, 1

PF[] : 10, 42, 48, 60, 80, 81

arr() = 10, 32, 6, 12, 20, 1

PF() : 10, 42, 48, 60, 80, 81

pf[0] = ar[0]

pf[[redacted]] = ar[0] + ar[1]

pf[2] = ar[0] + ar[1] + ar[2]

pf[3] = ar[0] + ar[1] + ar[2] + ar[3]

+

⋮

```
PF[N]
for(i=0; i<N; i++)
{
        PF[i]=0
        for(j=0; j≤i; j++)
        { PF[i] = PF[i] + ar[j]
}
```

PF is ready.

i	j	#
0	0-0	1
1	0-1	+ 2
2	0-2	+ 3
3	0-3	+ 4
4	0-4	+ 5
⋮		+
N-1	0-(N-1)	N

1 + 2 + 3 + 4 + . . . N

$$= \frac{N(N+1)}{2}$$

$$= \frac{N^2}{\cancel{2}} + \cancel{\frac{N}{2}}$$

TC : $O(N^2)$

PF[0] = ar[0]

PF[1] = PF[0] + ar[1]

PF[2] = PF[1] + ar[2]

PF(3) = PF(2) + ar[3]

PF[N] TC: O(N)

```
PF[0] = ar[0]
for( i=1 ; i<N ; i++)
{
    PF[i] = PF[i-1] + ar[i]
```

PF is ready

$$\begin{array}{cccccccccccc} & & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \text{Stock}[\,] & = & [-5, & 10, & 20, & 40, & 50, & -10, & 80, & -90, & -20, & -10] \end{array}$$

$$\begin{array}{cccccccccccc} \text{PF}[\,] & = & [-5, & 5, & 25, & 65, & \underline{115}, & 105, & 185, & 95, & 75, & 65] \\ & & \underline{0} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array}$$

Start day	End date	Net Profit / loss
1	4	120
0	0	-5
7	9	-120
0	9	65
2	7	90

$\rightarrow$ PF[4] - PF[0]
= 115 - (-5)
= 115 + 5 = 120

```
PF[N]
PF[0] = ar[0]
for ( i = 1 ; i < N ; i++)
{     PF[i] = PF[i-1] + stock[i]
```

```
for (each query)
{
    l, r
    if (l > 0)
    {
        print (PF[r] - PF[l-1])
    }
    else
        print (PF[r])
}
```

TC: $O(N + Q)$

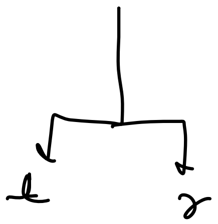
SC: $O(N)$

Stock[] =

0	1	2	3	4	5	6	7	8	9
-5	~~10~~	~~20~~	~~40~~	~~50~~	-10	80	-90	-20	-10
	5	25	65	115	...				

SC: $O(1)$

Q. Given array N size. Q queries,



For each query, return the sum of all even-indexed elements.

A[] = {2, 3, 1, 6, 4, 5}
0 1 2 3 4 5

Query:	ans
1 3	1
2 5	5
0 5	7

BF

For each query
iterate & get the sum

TC: O(QN)

$A[\,] = \{2, 3, 1, 6, 4, 5\}$

$\quad\;\; 0 \;\; 1 \;\; 2 \;\; 3 \;\; 4 \;\; 5$

$PF_e[\,] = \{2, 2, 3, 3, 7, 7\}$

$\quad\;\; 0 \;\; 1 \;\; 2 \;\; 3 \;\; 4 \;\; 5$

$1-5 = PF_e[5] - PF_e[0]$

$= 7 - 2$

$= 5$

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

use in last Q as well

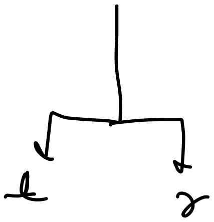
```
for ( each query )
{
        l, r
        if ( l > 0 )
        {
                print ( PF_e[r] - PF_e[l-1] )
        else
                print ( PF_e[r] )
}
```

Quiz $[2, 4, 3, 1, 5]$

$\ \ 0\ \ 1\ \ 2\ \ 3\ \ 4$

$PF_e : [2, 2, 5, 5, 10]$

Q. Given array N size. Q queries,



For each query, return the sum of all even-indexed elements.

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

Q Given an array of size N. Count how many special indexes are there.

Those indexes, which when removed from the array, then sum of all Even indexed element is equal to sum of all odd indexed element.

$Arr[] = \{4, 3, 2, 7, 6, -2\}$ → ans = 2
$\phantom{Arr[] = \{}0\ \ 1\ \ 2\ \ 3\ \ 4\ \ 5$

i	Arr	S_E	S_O	Special index
0	{3, 2, 7, 6, -2} 0 1 2 3 4	8	8	✓
1	{4, 2, 7, 6, -2} 0 1 2 3 4	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	✗

Quiz : $\{4, 1, 3, 7, 10\}$

$0 \quad 1 \quad 2 \quad 3 \quad 4$

Sum of odd (After 2 is removed) = $\{4, 1, 7, 10\}$

$0 \quad 1 \quad 2 \quad 3$

= 11

Quiz $\{2, 3, 1, 4, 0, -1, 2, -2, 10, 8\}$

Sum of odd (After 3 is removed) : $\{2, 3, 1, 0, -1, 2, -2, 10, 8\}$

= $3 + 0 + 2 + 10$

= 15

Sum of Even (After 3 is removed) : $\{2, 3, 1, 0, -1, 2, -2, 10, 8\}$

= $2 + 1 - 1 - 2 + 8$

= 8

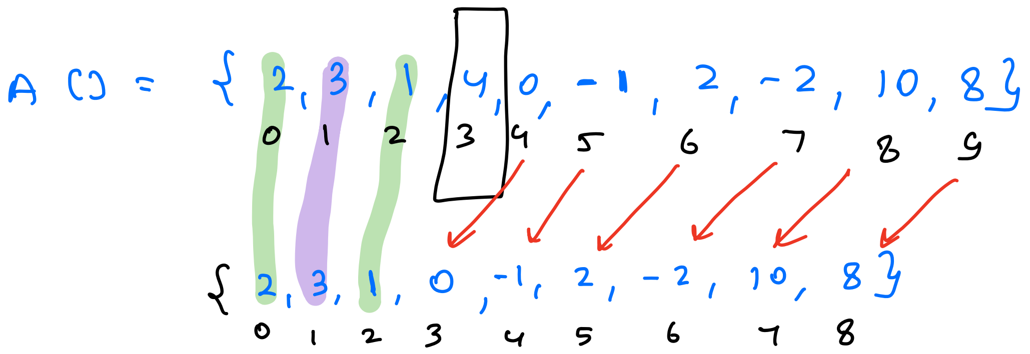
BF

Delete N

Calculate N + N

Undo N

TC: $O(N^2)$



S_o [all] in new array = S_o [left of deletion] old + S_E [right of deletion] old

S_E [all] in new array = S_E [left] old + S_o [right] old

ans = 0

(l, r)

$PF_o[r] - PF_o[l-1]$

```
for ( i=0 ; i< N ; i++ )
{
    $x = PF_o[i-1]$
    $y = PF_E[N-1] - PF_E[i]$
    $x' = PF_E[i-1]$
    $y' = PF_o[N-1] - PF_o[i]$

    $S_o\ new = \overset{x}{S_o(0,i-1)} + \overset{y}{S_E(i+1, N-1)}$
    $S_e\ new = \underset{x'}{S_e(0,i-1)} + \underset{y'}{S_o(i+1, N-1)}$

    if ( $S_o$new == $S_e$new)   ans++
}

return ans
```

```
// Calculate $PF_o$
// calculate $PF_E$

$i = 0$
$a = S_o = S_E$ (right) $= S_E(i+1, N-1) = PF_E(N-1] - PF_E[i]$
$b = S_e = S_o$ (right) $= S_o(i+1, N-1) = PF_o[N-1] - PF_o[i]$
    if( a == b )   ans++

$i == N-1$
$a = S_o = S_o$(left) $= S_o(0, i-1) = PF_o[i-1]$
$b = S_e = S_e$ (left) $= S_e(0, i-1) = PF_e[i-1]$
    if( a == b )   ans++

for( i = 1 ; i < N-1, i++ )
{
    $S_o$ new = $PF_o[i-1] + PF_E[N-1] - PF_E[i]$
    $S_e$ new = $PF_E[i-1] + PF_o[N-1] - PF_o[i]$

    if( $S_o$new == $S_e$new )   ans++
}

return ans
```

TC: N+N+N

TC: O(N)

SC: N+N

SC: O(N)

$$S_o\,new = S_o(left) + S_E(right)$$

$$= S_o(0, i-1) + S_E(\underline{i+1}, N-1)$$

$$= PF_o[i-1] + PF_E[N-1] - PF_E[i]$$

```
ans=0
    // i==0
if ($PF_E[N-1] - PF_E[0]$ == $PF_o[N-1] - PF_o[0]$)    ans++
    // i== N-1
if( $PF_o[N-2]$ == $PF_E[N-2]$    ans++

for ( i= 1 ; i< N-1, i++ )
{
    $S_o$ new = $PF_o[i-1] + PF_E[N-1] - PF_E[i]$
    $S_e$ new = $PF_E[i-1] + PF_o[N-1] - PF_o[i]$

    if ( $S_o$new == $S_e$new)    ans++

return ans
```

$Arr[\,] = \{4, 3, \cancel{2}, 7, 6, -2\}$ → $\{4, 3, 7, 6, -2\}$

$\quad\quad\quad\quad\;\; 0 \quad 1 \quad \cancel{2} \quad 3 \quad 4 \quad 5 \qquad\qquad\quad 0 \quad 1 \quad 2 \quad 3 \quad 4$

$PF_o[\,] = \{0, 3, 3, 10, 10, 8\}$

$PF_E[\,] = \{4, 4, 6, 6, 12, 12\}$

$i = 2$

$S_o = S_o(\text{left}) + S_E(\text{right})$

$= S_o(0,1) + S_E(3,5)$

$= 3 + 12 - 6$

$= 9$

$S_E = S_E(\text{left}) + S_o(\text{right})$

$= 4 + 8 - 2$

$= 9$