Q.1 Given an Array. Create and return prefix sum Array.

where $ps[i] = A[0] + A[1] + A[2] + \ldots\ldots + A[i]$.

$$\begin{array}{c|cccccc} A = & [2 & 4 & 5 & -3 & 17 & 8] \\ & 0 & 1 & 2 & 3 & 4 & 5 \end{array}$$

$$\begin{array}{c|cccccc} ps = & [2 & 6 & 11 & 8 & 25 & 33] \\ & 0 & 1 & 2 & 3 & 4 & 5 \end{array}$$

$$\begin{array}{c|ccccc} A = & [2 & 9 & -3 & 5 & 1] \\ & 0 & 1 & 2 & 3 & 4 \end{array}$$

$$\begin{array}{c|ccccc} ps = & [2 & 11 & 8 & 13 & 14] \\ & 0 & 1 & 2 & 3 & 4 \end{array}$$

Observations

$ps[0] = A[0]$

$ps[1] = \underbrace{A[0]}_{ps[0]} + A[1] \Rightarrow ps[0] + A[1]$

$ps[2] = \underbrace{A[0] + A[1]}_{ps[1]} + A[2] \Rightarrow ps[1] + A[2]$

$ps[3] = \underbrace{A[0] + A[1] + A[2]}_{ps[2]} + A[3] \Rightarrow ps[2] + A[3]$

$$\boxed{ps[i] = ps[i-1] + A[i]}$$

$$A = \begin{array}{cccccc} [\,3 & 9 & 4 & -5 & 2\,] \\ 0 & 1 & 2 & 3 & 4 \end{array}$$

$$ps = \begin{array}{cccccc} [\,3 & 12 & 16 & 11 & 13\,] \\ 0 & 1 & 2 & 3 & 4 \end{array}$$

```
int[] prefixSum(int[] A) {
        int n= A.length;
        int[] ps= new int [n];
        ps[0]= A[0];
        for(int i= 1; i<n; i++) {
                ps[i] = ps[i-1] + A[i];
        }
        return ps;
}
```

$$A = \begin{array}{ccccc} [\,3 & 9 & 4 & -5\,] \\ 0 & 1 & 2 & 3 \end{array}$$

$$ps = \begin{array}{ccccc} [\,3 & 12 & 16 & 11\,] \\ 0 & 1 & 2 & 3 \end{array}$$

i	
1	ps[1] = ps[0]+A[1]
2	ps[2] = ps[1]+A[2]
3	ps[3] = ps[2]+A[3]

Q.2 Range Sum queries

Given an array and Q queries. Find the answer for all queries in the given range.

$1 <= n <= 10^5$

$1 <= Q <= 10^5$

A = [3 4 -2 6 8 10 13 1]
0 1 2 3 4 5 6 7

Q = 4 (L <= R)

L	R	ans
1	3	8
2	6	35
5	5	10
0	3	11

① . brute force

→ go on each query find L and R from that and then find sum of array from L to R.

```
void   solve  (int [] A, int [][] Q) {

    for (int i=0; i< Q.length; i++) {
        int L= Q[i][0];
        int R= Q[i][1];
        // find sum of A[] from L to R
        int sum= 0;
        for (int k= L; k<=R; k++) {
                sum += A[k];
        }
        SOPln(sum);
    }
}
```

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

```
Q = [ [0, 3]
      [1, 4]
    ]
```

i=0, L=0, R=3, sum=10

i=1, L=1, R=4, sum=8

TC : O(Q*N)

TLE

Improvements

$$A = \begin{array}{cccccccc} [3 & 4 & -2 & 6 & 8 & 10 & 13 & 1] \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array}$$

$$PS = \begin{array}{cccccccc} [3 & 7 & 5 & 11 & 19 & 29 & 42 & 43] \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array}$$

$\text{sum}(3,6) = \text{sum}(0,6) - \text{sum}(0,2)$

$PS[6] - PS[2] = 42 - 5 = 37$

$\text{sum}(2,5) = \text{sum}(0,5) - \text{sum}(0,1)$

$PS[5] - PS[1] = 29 - 7 = 22$

$\text{sum}(0,3) = PS[3]$

*** =

$$\boxed{\text{sum}(L,R) = PS[R] - PS[L-1]}$$

$L != 0$

```
sum (L to R)

    if (L == 0) {
            sop (ps[R]);
    }
    else {
            sop (ps[R] - ps[L-1]);

    }


void solve (int[] A, int[][] Q) {
    int[] ps = prefixSum(A);

    for (int i=0; i< Q.length; i++) {
        int L= Q[i][0];
        int R= Q[i][1];
        // find sum of A[] from L to R
        if (L == 0) {
                sop (ps[R]);
        }
        else {
                sop (ps[R] - ps[L-1]);
        }
    }
}
```

TC: O(N+Q)

SC: O(N)

Q.3 Equilibrium Index

Amazon, Adobe

Given an Array, find the equilibrium index.

i is an equilibrium index when :

sum of all elements on the left side of i = sum of all elements on the right side of i

$$\begin{array}{l|ccccccc}
A: & -7 & 5 & 1 & \boxed{2} & -4 & 3 & 0 \\
 & 0 & 1 & 2 & \boxed{3} & 4 & 5 & 6
\end{array}$$

ans = 3

$$\begin{array}{l|ccccc}
A: & 5 & 1 & 3 & \boxed{2} & 9 \\
 & 0 & 1 & 2 & \boxed{3} & 4
\end{array}$$

ans = 3

$$\begin{array}{l|ccc}
A: & 1 & 2 & 3 \\
 & 0 & 1 & 2
\end{array}$$

ans = -1

brute force : go on every index

→ find ls (0 to i-1)

→ find rs (i+1 to n-1)

→ compare ls and rs

```
for ( i ) {
    for (0 to i-1) : ls
    for (i+1 to n-1) : rs
    if (ls == rs) return i;
}
```

optimised idea

sum of all elements on the left side of i = sum of all elements on the right side of i

sum(0, i-1) = sum(i+1, n-1)

ps[i-1] = ps[n-1] - ps[i]

```
int equilibriumIndex ( int[] A) {

    int[] ps = prefixSum(A);
    int n= A.length;

    for (int i=0; i<n; i++) {
         int ls = 0;
         if ( i>0) {
             ls = ps[i-1];
         }
         int rs = ps[n-1] - ps[i];
         if(ls == rs) {
             return i;
         }
    }

    return -1;
}
```

A: 5 1 3 2 9
   0 1 2 3 4

ps: 5 6 9 11 20
    0 1 2 3 4

i	ls	rs
0	0	20-5 = 15
1	5	20-6 = 14
2	6	20-9 = 11
3	9	20-11 = 9

TC : O(N)

Q.4 Given an Array and Q queries, find the count of even numbers for every query.

$$\begin{array}{l|cccccccc} A: & 3 & 5 & 8 & 9 & 16 & 14 & 13 & 12 \\ & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array}$$

Queries

L	R	ans
1	5	3
2	6	3
4	5	2
4	4	1
3	6	2

i) how to solve $O(Q*N)$ {hw}

ii) Improvise

$$\begin{array}{l|cccccccc} A: & 3 & 5 & 8 & 9 & 16 & 14 & 13 & 12 \\ & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array}$$

$$\begin{array}{l|cccccccc} pc: & 0 & 0 & 1 & 1 & 2 & 3 & 3 & 4 \\ & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array}$$

↓
prefix count

pc[i] ⇒ no. of even element in A[] from 0 to i

$$\begin{array}{lcccccccc} A: & [3 & 5 & 8 & 9 & 16 & 14 & 13 & 12] \\ & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array}$$

$$\begin{array}{lcccccccc} pc: & [0 & 0 & 1 & 1 & 2 & 3 & 3 & 4] \\ & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array}$$

$(2, 5) \longrightarrow pc[5] - pc[1] = 3 - 0 = 3$

$(4, 7) \longrightarrow pc[7] - pc[3] = 4 - 1 = 3$

```
void  solve ( int [] A, int [] [] Q ) {
        int [] pc = prefix Count (A);

        for(int  i=0; i< Q.length; i++ ) {
            int  L= Q [i] [0];
            int  R= Q [i] [1];

            if (L==0) {
                SOP (PC[R]);
            }
            else {
                SOP ((PC[R] - PC[L-1]));
            }
        }
}
```

TC: O(N+Q)

SC: O(N)

$$\begin{array}{lccccccc} A = & [& 3 & 4 & 9 & 8 & 5 &] \\ & & 0 & 1 & 2 & 3 & 4 & \end{array}$$

$$\begin{array}{lccccccc} pc = & [& 0 & 1 & 1 & 2 & 2 &] \\ & & 0 & 1 & 2 & 3 & 4 & \end{array}$$

$$\underset{\downarrow}{pc[i]} = pc[i-1] + \underset{\text{based on } A[i]}{(1 \text{ or } 0)}$$

no. of even elements
from 0 to i

```
int[] prefixCount (int[] A) {

        int n= A.length;

        int[] pc= new int[n];

        if (A[0]%2 == 0) {
              pc[0] = 1;
        }
        else {
              pc[0] = 0;
        }

        for (int i=1; i<n; i++) {
              int temp=0;
              if(A[i]%2 == 0) {
                     temp=1;
              }

              pc[i]= pc[i-1] + temp;
        }

        return pc;
}
```

$$A = \begin{array}{cccccc} [2 & 5 & 4 & 6 & 7 & 8] \\ 0 & 1 & 2 & 3 & 4 & 5 \end{array}$$

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

Doubts
=

i is equilibrium index

sum of ele on left of i = sum of ele on right of i

sum(0, i-1) $\Downarrow$ ps[i-1]

sum(i+1, n-1) $\Downarrow$ ps[n-1] - ps[i]

A :	2	1	4	3
	0	1	2	3

ps:	2	3	7	10
	0	1	2	3

i	ls	rs
0	0	10-2=8
1	2	10-3=7
2	3	10-7=3

Pick from both sides

carry forward

B = 4

$$\begin{array}{cccccccc} [2 & 4 & 5 & 6 & 7 & 3 & 1 & 9] \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array}$$

left	right
4	0
3	1
2	2
1	3
0	4

$$\begin{array}{cccccccc} & & & i & & & & j \\ [2 & 4 & 5 & 6 & 7 & 3 & 1 & 9] \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array}$$

B = 4

i = B-1

j = n-1

sum = 2+4+5+6 , ans = sum (initialse)

```
{ sum -= A[i]
  sum += A[j]
  i--;
  j--;
```

i	j	sum
3	7	$2+4+5+\cancel{6}-\cancel{6}+9$
2	6	$2+4+\cancel{5}+9-\cancel{5}+1$
1	5	$2+\cancel{4}+9+1-\cancel{4}+3$
0	4	$\cancel{2}+9+1+3-\cancel{2}+7$

a g g a g a a g

cnt = ~~1~~ ~~2~~ ~~3~~ 4

ans = 1 + 1 + 2 + 4

```
for(int i=0; i<=N; i++) {
        for(int j=i; j<=N && j>i; j++) {
                sop();
        }
}
```

false (j is starting from i)
j = i

O(N)