Q1) Given N array elements, print prefix array (pf)

Where $pf[i] = sum(A[0:i])$

	0	1	2	3	4
A = {	3	2	-1	5	2 }
pf = {	3	5	4	9	11 }

$pf[0] = sum(A[0:0])$

$pf[1] = sum(A[0:1])$

$pf[2] = sum(A[0:2])$

	0	1	2	3	4	5	6
A = {	5	2	-1	0	4	3	2 }
Ans: {	5	7	6	6	10	13	15 }

A: [10   32   6   12   20   1]

= { 10   42   48   60   80   81 }

$$j \qquad\qquad i$$
$$0 \quad 1 \quad 2 \quad 3 \quad 4$$

$$A = \{\ 3 \quad 2 \quad -1 \quad 5 \quad 2\ \}$$

$$pf[3] = A[0] + A[1] + A[2] + A[3]$$

Normal

```
pf = new Array [N]
for (i = 0; i < N; i++) {

    sum = 0
    for (j = 0; j <= i; j++) {
        sum += A[j]
    }
    3

    pf[i] = sum
}

print (pf)
```

pf[i]

TC: $O(N^2)$

SC: $O(N)$

$pf[0] = sum(A[0:0]) = A[0]$

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

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

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

$pf[i] = sum(A[0:i]) = \underbrace{sum(A[0:i-1])}_{pf[i-1]} + A[i]$

$$pf[i] = pf[i-1] + A[i] \quad \forall \ i > 0$$

$$pf[0] = A[0]$$

$pf = new \ Array[N]$

$pf[0] = A[0]$

TC: $O(N)$

SC: $O(N)$

```
for (i = 1; i < N; i++) {
     pf[i] = pf[i-1] + A[i]
}
```

HW:

SC: $O(1)$

use the same

array

print (pf)

Q2) Given N array elements & Q queries on same array
For each query calculate sum of all elements in range.

$$A = \{ \overset{0}{-3} \quad \overset{1}{6} \quad \overset{2}{2} \quad \overset{3}{4} \quad \boxed{\overset{4}{5} \quad \overset{5}{2} \quad \overset{6}{8} \quad \overset{7}{-9} \quad \overset{8}{3}} \quad \overset{9}{13} \} \; N$$

L = { 4 3 1 0 6 7 }
R = { 8 7 3 4 9 7 }

Q = 7

find Range Sum (A, L, R) {

L ≤ R

L	R	
4	8	→ 9
3	7	→ 10
1	3	→ 12
0	4	→ 14
6	9	→ 3
7	7	→ -9

$$\overset{Q}{\text{for }} (i=0; i < \text{len}(L); i++) \{$$

sI = L[i]   TC: O(QN)

eI = R[i]   SC: O(1)

sum = 0

for (j = si; j <= eI; j++) {

sum += A[j]

}

print(sum)

}

	0	1	2	3	4	5	6	7	8	9
A = {	-3	6	2	4	5	2	8	-9	3	1 }

	0	1	2	3	4	5	6	7	8	9
pf = {	-3	3	5	9	14	16	24	15	18	19 }

$$\text{sum}(A[4:8]) = pf[8] - pf[3]$$

$$pf[8] = \text{sum}(A[0:8])$$

$$pf[8] = \underbrace{\text{sum}(A[0:3])}_{pf[3]} + \text{sum}(A[4:8])$$

$$pf[8] = pf[3] + \text{sum}(A[4:8])$$

$$pf[8] - pf[3] = \text{sum}(A[4:8])$$

*****

$$\text{sum}(A[l:r]) = pf[r] - pf[l-1] \quad \forall \quad l > 0$$

$$\text{sum}(A[0:r]) = pf[r] \qquad l = 0 \text{ is edge case}$$

$$sum(A[l:r])$$

$$pf[r] = sum(A[0:r])$$
$$pf[r] = sum(A[0:l-1]) + sum(A[l:r])$$
$$pf[r] = pf[l-1] + sum(A[l:r])$$

*****

$$sum(A[l:r]) = pf[r] - pf[l-1] \quad \forall \quad l > 0$$
$$sum(A[0:r]) = pf[r] \quad l = 0 \text{ is edge case}$$

1) find pf array

2) for $(i=0; i < len(L); i++)$ {

    Q

    $sI = L[i]$

    $lI = R[i]$ $A[si:ei]$

    if $(si == 0)$ {   sum = $pf[ei]$ }

    else {

    | sum = $pf[ei] - pf[si-1]$

    }

    print(sum)

}

TC: $O(N+Q)$

SC: $O(N)$

pf array

Break (10:21 - 10:31)

Microsoft

Q3) Equilibrium index

Given N array elements, count no. of equilibrium index

i^{th} element is equilibrium if

Sum of all elements = Sum of all elements

before i^{th} index   after i^{th} index

[0, i-1]   [i+1, N-1]

Ex1)

	0	1	2	3
A[4] =	3	2	4	-1
before :	0	3	5	9
after :	5	3	-1	0

Ans : 1

count of eq indexes

Ex2)

	0	1	2	3	4	5	6
A[] =	-7	1	5	2	-4	3	0
before:	0	-7	-6	-1	1	-3	0
after:	7	6	5	-1	3	0	0

Ans : 2

```
count = 0
for (i=0; i<N; i++) {
        before =    sum (A[0:i-1])          ⎫   2 for loops
        after  =    sum (A[i+1 : N-1])      ⎭      TC: O(N²)
        if (before == after) {
        |
        |           count ++
        }
}
return   Count
```

$$sum (A[l:r]) =   pf[r] - pf[l-1] \quad \forall \quad l > 0$$

$$l = i+1 \qquad r = N-1$$

1)   Find   pf array   TC: O(N)

2)   count = 0   SC: O(N)
                                   pf array
```
for (i=0; i<N; i++) {
        if (i==0) {before = 0} else { before = pf[i-1]}
        after =    sum (A[i+1 : N-1]) ⟶ pf[N-1] - pf[i]
        if (before == after) {
        |
        |           count ++
        }
}
return   Count
```

At i=0   we have an edge
                                   case

Q4) Given N array elements find pf_even of size N
(Sum of all even index elements from [0-i])

A[7] = { 2   3   4   2   -1   3   5 }

pf even = { 2   2   6   6   5   5   10 }

A[8] = { 2   1   1   3   -1   5   8   2   1   6 }

{ 2   2   3   3   2   2   10   10   11   11 }

Solution!

$$pf[0] = A[0]$$

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

curr index: pf[i] = pf[i-1] + A[i]

i%2 == 0

odd index: pf[i] = pf[i-1]

else

}

```
        0    1    2    3    4    5    6
{   2    3    4    2   -1    3    5 }

pfodd {  0    3    3    5    5    8    8 }
```
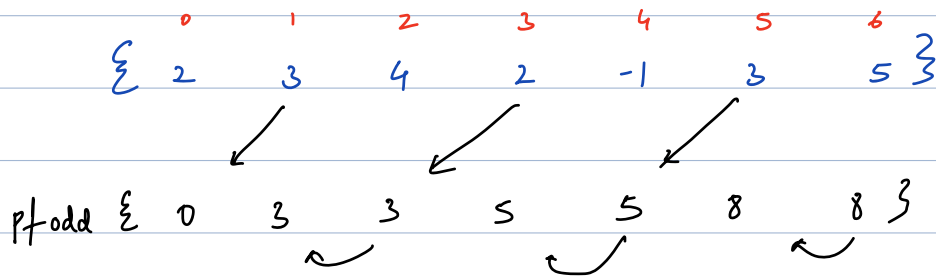
$pfodd[0] = 0$ // why?   0 is even index

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

even index: $pf[i]= pf[i-1]$
$\overline{i\%2==0}$

odd index: $pf[i]= pf[i-1] + A[i]$
$\overline{else}$
$\}$

HW: Try
                    even index   sum   for $[l:r]$

```
                          odd        even
              0    1    2    3    4    5    6
{   2    3    4    2   -1    3    5 }
```
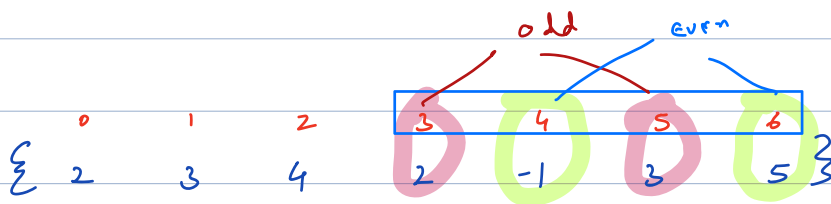
even index sum $[3:6]$: 4 ans

Try $pfeven[r] - pfeven[l-1]$?

Done!