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

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

$$
\begin{array}{cccccc}
 & 0 & 1 & 2 & 3 & 4 \\
A = \{ & 3 & 2 & -1 & 5 & 2 \} \\
pf = \{ & 3 & 5 & 4 & 9 & 11 \}
\end{array}
$$

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

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

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

$$
\begin{array}{cccccccc}
 & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\
A = \{ & 5 & 2 & -1 & 0 & 4 & 3 & 2 \} \\
Ans = \{ & 5 & 7 & 6 & 6 & 10 & 13 & 15 \}
\end{array}
$$

$$
\begin{array}{ccccccc}
A: [& 10 & 32 & 6 & 12 & 20 & 1] \\
= \{ & 10 & 42 & 48 & 60 & 80 & 81 \}
\end{array}
$$

j i

0 1 2 3 4

A = { 3 2 -1 5 2 }

Normal

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

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

print (pf)
```

pf[i]

TC: $O(N^2)$

SC: $O(N)$

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

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

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

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

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

$$\boxed{\begin{aligned} pf[i] &= pf[i-1] + A[i] \quad \forall\ i > 0 \\ pf[0] &= A[0] \end{aligned}}$$

```
pf = new Array(N)
pf[0] = A[0]

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

print(pf)
```

TC: O(N)

SC: O(N)

HW:

SC: O(1)

use the same array

Q2) Given N array elements & Q queries on same array

For each query calculate sum of all elements in range.

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

N

L = { 9 3 1 0 6 7 }

R = { 8 7 3 4 9 7 }

Q = 7

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

```
findRangeSum (A, L, R){

for (i=0; i< len(L); i++){        // len(L) = Q
    sI = L[i]
    eI = R[i]
    sum = 0
    for (j= si ; j<=eI ; j++){
        sum += A[j]
    }

    print(sum)
}
```

TC: $O(QN)$

SC: $O(1)$

$$\begin{array}{lcccccccccc} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ A = \{ & -3 & 6 & 2 & 4 & 5 & 2 & 8 & -9 & 3 & 1 \} \end{array}$$

$$\begin{array}{lcccccccccc} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ pf = \{ & -3 & 3 & 5 & 9 & 14 & 16 & 24 & 15 & 18 & 19 \} \end{array}$$

$$sum(A[4:8]) = pf[8] - pf[3]$$

$$pf[8] = sum(A[0:8])$$

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

$$pf[8] = pf[3] + sum(A[4:8])$$

$$pf[8] - pf[3] = sum(A[4:8])$$

* * * * *

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

$sum(A[l:r])$

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

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

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

*****

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

$sum(A[0:r]) = pf[r] \quad l=0$ is edge case

1) find pf array

2) for (i=0 ; i < $\underbrace{len(L)}_{Q}$; i++) {

  sI = L[i]

  eI = R[i]   A[si : ei]

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

  else {

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

  }

  print(sum)

}

TC: O(N+Q)

$\boxed{SC: O(N)}$

pf array

Break (10:21 - 10:31)

Microsoft

Q3) Equillibrium index

Given N array elements, count no. of equillibrium index

i^{th} element is equillibrium if

Sum of all elements before i^{th} index $[0, i-1]$ = Sum of all elements after i^{th} index $[i+1, N-1]$

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])
    after = sum(A[i+1 : N-1])
    if (before == after) {
        count++
    }
}
return count
```

} 2 for loops

TC: O(N^2)

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

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

1) Find pf array — TC: O(N)

$\boxed{SC: O(N)}$ pf array

2)
```
count = 0
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

1

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

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

	0	1	2	3	4	5	6	7	8	9
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++){
even index :  pf[i]= pf[i-1] + A[i]
i%2 ==0
odd index :   pf[i] = pf[i-1]
else
}
```

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

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]
    i%2 == 0
    odd index : pf[i] = pf[i-1] + A[i]
    else
}
```

HW: Try

even index sum for [l:r]

odd even

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

even index sum [3 : 6] : 4 ans

Try pfeven[R] - pfeven[L-1] ?

Done!