Question :

Given an array representing profit or loss from a stock over a period of days, write a function that calculates total profit or loss for a given range of days.

Example :

	0	1	2	3	4	5	6	7	8	9
SP[] :	-5	10	20	40	50	-10	80	-90	-20	-10
C :	-5	5	25	65	115	105	185	95	75	65

Queries

S	E	Net	
0	9	65	N
1	4	120	N
0	0	-5	N
7	9	-120	N
2	7	90	N

Q queries.

O(QN)

```
int [][] queries = new int [Q][2];
```

queries [i] → ith query.
queries [i][0] → s for ith query.
queries [i][1] → e for ith query.

```
void printQuerySum(int[] A, int[][] Q)
{
    for(i=0; i<Q.length; i++)
    {
        int[] query = Q[i];
        int s = query[0];
        int e = query[1];
        int sum = 0;
        for(j=s; j<=e; j++)
        {
            sum = sum + A[j];
        }
        print(sum);
    }
}
```

TC : O(N*Q)

SC : O(1)

Question:

Given N elements and Q queries.
For each query, calculate sum of all elements from L to R.

Eg:

	0	1	2	3	4	5	6	7	8	9
A :	-3	6	2	4	5	2	8	-9	3	1

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

Given the scores of the 1st 10 overs of a cricket match

	1	2	3	4	5	6	7	8	9	10
	2	6	6	15	2	18	16	14	9	9
	2	8	14	29	31	49	65	79	88	97

a. Runs scored in 7th over = T(7) - T(6)
= 65 - 49 = 16

b. Runs scored from 6th to 10th over
S(6-10) : T(10) - T(5)

c. Runs scored in the 10th over :
S(10) - T(10) - T(9)

s	e	Runs
3	6	T(6) - T(2)
4	9	T(9) - T(3)
s	e	T(e) - T(s-1)

Prefix Sum Array:

	0	1	2	3	4
A:	2	5	-1	7	1
PF:	2	7	6	13	14

	0	1	2	3	4	5
A:	10	32	6	12	20	1
PF:	10	42	48	60	80	81

	0	1	2	3	4	5	6	7	8	9
A:	-3	6	2	4	5	2	8	-9	3	1

L	R	Sum
4	8	
3	7	
1	3	
0	4	
7	7	

Creati a prefix array:

	0	1	2	3	4	5	6	7	8	9
SP[] :	-5	10	20	40	50	-10	80	-90	-20	-10

```
long[]  getPrefixSum (int[] A)
{
    long[] psa = new long[A.length];
    psa[0] = A[0];
    for(i=1 ; i<A.length ; i++)
    {
        psa[i] = psa[i-1] + A[i]
    }
   return psa ;
}
```

Using prefix array to answer queries:

```
void printQuerySum(int[] A, int[][] Q)
{
    long[] pa = getPrefixSum(A);
    for (i=0; i<Q.length; i++)
    {
        int[] query = Q[i];
        int s = query[0];
        int e = query[1];
        if(s==0)
        { print ps[e]; }
        else
        {print (pa[e] - pa[s-1]); }
    }
}
```

TC: O(N+Q)

SC: O(N)

Break : 10:52 PM

Question :

Given an array of size N and Q queries with start and end index. For each query return the sum of all even indexed elements from s to e.

Brute Force O(N*Q)

		*		*		*	
		0	1	2	3	4	5
A :		2	3	1	6	4	5
pse :		2	2	3	3	7	7

s	e	sum	P[E] - P[S-1]
1	3	1	P(3) - P(0) : 1
2	5	5	P(5) - P(1) : 5
0	4	7	P(4) : 7
3	3	0	P(3) - P(2) : 0

```
long[] getPrefixSumEv (int[] A)
{
    long[] psa = new long[A.length];
    psa[0] = A[0];
    for(i=1 ; i<A.length ; i++)
    {  if(i%2==0)
       { psa[i] = psa[i-1] + A[i];}
       else
       { psa[i] = psa[i-1] ;}
    }
    return psa ;
}
```

HW for odd

Special Index:

Given A, calculate the no. of special indices.

	0	1	2	3	4	5
A :	4	3	2	7	6	-2

	0	1	2	3	4	5
S_O :	8	8	9	9	10	10
S_E :	8	9	9	4	4	12

	0	1	2	3	4	5
A :	4	3	2	7	6	-2
pse :	4	4	6	6	12	12
pso :	0	3	3	10	10	8

HW

$S_{O(3)}$: Sum of all odd index [0-2] + Sum of all even index [4-5]

$S_{E(3)}$: Sum of all even index [0-2] + Sum of all odd index [4-5]

0	1	2	3	4	5	6	7	8	9
2	3	1	4	0	-1	2	-2	10	8

Carry Fwd Subarrays.

```
int countSpecial(int[] A)
{
  int count = 0;
  int[] pse = getPrefixSumEv(A);
  int[] pso = getPrefixSumOdd(A);
  int N = A.length;
  for(i=0; i<A.length; i++)
  { int so=0; int se=0;
    if(i==0)
    {
        so = pse[N-1]          // Sum of even index
             - pse[0];           elements from
        se = pso[N-1]            1 → N-1
             - pso[0];
    }
    else
    {
        so = pso[i-1] +        // So[0-(i-1)]
             pse[n-1] - pse[i]; // + SE[i+1 → n-1]

        se = pse[i-1] +        // SE[0→(i-1)]
             pso[n-1] - pso[i]; //+ So[i+1 → n-1]
    }
    if(so==se)
    { count++; }
}
return count;
```

4

Doubts:

0	1	2	3	4
2	4	5	6	_

Q[ ] [ ] ⟶ fun( )