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

Queries

S	E	Net		
0	9	65	O(N)	O(Q*N)
1	4	120	O(N)	
0	0	-5	"	$1 \leq Q \leq 10^5$
7	9	-120	"	$1 \leq N \leq 10^5$
2	7	90	"	

```
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 s = Q[i][0];
        int e = Q[i][1];
        int sum=0;
        for (j=s; j<=e; j++)
        {
            sum = sum + A[j];
        }
        println(sum);
    }
}
```

$1 \le Q \le 10^5$

$1 \le N \le 10^5$

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
C :	-3	3	5	9	14	16	24	15	18	19

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

$C[R] - C[L-1]$

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

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

a. Runs scored in 7th over = $R(7) - R(6)$
$= \underline{16}$

b. Runs scored from 6th to 10th over = $R(10) - R(5)$
$= 97 - 31 = 66$

c. Runs scored in the 10th over : $R(10) - R(9) = 9$

d. Runs scored from 3rd to 6th over : $R(6) - R(2)$
$= \underline{41}$

Prefix Sum Array:

HW

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

PF:

	0	1	2	3	4	5
A:	10	32	6	12	20	1

PF:

	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
psa[] :	-5	5								

psa[1] = A[1] + psa[0]
psa[2] = A[2] + psa[1]
psa[i] = A[i] + psa[i-1]

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

    }
    return psa;
}
```

TC: O(N)

SC: O(N)

Using prefix array to answer queries:

```
void printQuerySum(int[] A, int[][] Q)
{ int[] psa = createPSA(A);          → O(N)
  for(i=0; i<Q.length; i++)
  { int s = Q[i][0];                 O(Q)
    int e = Q[i][1];
    if(s==0)                         TC: O(N+Q)
    { print(psa[e]);
    }                                SC: O(N)
    else { psa[e] - psa[s-1] };
  }
}
```

$1 \le Q \le 10^5$

$1 \le N \le 10^5$

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.

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

s	e	sum
1	3	1
2	5	5
0	4	7
3	3	0

pse[e] - pse[s-1]

	* 0	1	* 2	3	* 4
	2	4	3	1	5
pse:	2	2	5	5	10

```
long[] getPrefixSumEv (int[] A)
{
```

HW

Break 8:38 AM

```
}
```

Question: Given a string s of lower case letters, return the count of pairs (i, j) such that i < j and s[i] = 'a' and s[j] = 'g'.

Eg:

s = "a b e g a g"
(0, 3) (4, 5)
(0, 5) (4, 3)
3

s = "a c g d g a g"
(0, 2) (0, 6)
(0, 4) (5, 6)

s = "b c a g g a a g"
(2, 3) (5, 7)
(2, 4) (6, 7)
(2, 7)

Brute Force:

res :

```
function countAG( String s)
{  result =0;
   for (i=0 ; i< s.length ; i++)
   {  if( s[i] == 'a')
      {
         for (j= i+1 → s.length-1)
         {  if (s[j] == 'g')
            {  res ++;
            }
         }
      }
   }
   return res;
}
```

i	j	res.
0		0
1		0
2	[3, 7]	3
3		3
4		3
5	[6, 7]	4
6	[7, 7]	5

TC: $O(N^2)$

SC: $O(1)$

Optimised Solution:

Carry Fwd

	0	1	2	3	4	5	6	7
S :	b	c	a	g	g	a	a	g
								i

countA: ~~0 1 2~~ 3

res : ~~0 1 2~~ 5

TC : O(N)

SC : O(1)

```
if (g)
{ res = res + countA }
```

```
if (A)
{ countA++; }
```

Subarrays:

Continuoces part of an array.

	0	1	2	3	4	5	6	7	8
A:	4	1	2	3	-1	6	9	8	12

2, 3, -1, 6 → ✓

9 → ✓

4, 12

1, 2, 6

3 2 1 4

2 4 1 6 -3 7 8 4

s = 0 e → n

1, 6, 8

7 8 4

(s, e)

s	ne	N
0	N	N
1	N-1	N-1
5	N-5	N-5

(4, 6) → -1, 6, 9

(0, 4) → 4, 1, 2, 3, -1

Representation Of A Sub Array:

	0	1	2	3	4	5	6	7	8
A:	4	1	2	3	-1	6	9	8	12

(start, end)

i → N-i

7 - 1 ⇒ 6.

Problem: Print all sub-arrays of a given array A.

	0	1	2
A:	3	8	10

i	j	
0	[0-2]	→ 3
1	[1-2]	→ 2
2	[2-2]	→ 1

```
for (i → 0 to N-1)   // fix s
{
    for (j → i to N-1)   // fix e
    {
        for (k → i to j)
        {

        }
    }
}
```

N^2 ↳ N

$O(N^3)$

Prefix sum
Carry Forward
Subarrays.