Q → Given an integer array,
find the total sum of all possible subarrays.

$A = [\overset{0}{3} \quad \overset{1}{2} \quad \overset{2}{5}]$

3 → 3
3 2 → 5
3 2 5 → 10
2 → 2
2 5 → 7
5 → $\underline{5}$

$\underline{32}$ (Ans)

$A = [\overset{0}{2} \quad \overset{1}{7}]$

2 → 2
2 7 → 9
7 → $\underline{7}$

$\underline{18}$ (Ans)

Bruteforce → ∀ subarrays, calculate sum & add it in the answer.

```
ans = 0
for L → 0 to (N-1) {                 L — R
    for R → L to (N-1) {
        sum = 0
        for i → L to R {             subarray sum L — R
            sum += A[i]
        }
        ans += sum                   TC = O(N^3)
    }                                SC = O(1)
} return ans
```

Prefix Sum →

```
P[0] = A[0]
for i → 1 to (N-1) {
    P[i] = P[i-1] + A[i]
}

ans = 0
for L → 0 to (N-1) {                L — R
    for R → L to (N-1) {
        if (L == 0)   ans += P[R]
        else          ans += P[R] - P[L-1]
    }
} return ans
```

$TC = O(N + N^2) = \underline{O(N^2)}$ $SC = \underline{O(N)}$

$A = [\overset{0}{3}\ \ \overset{1}{2}\ \ \overset{2}{5}]$

s =

L = 0 { $\overset{0 — 0}{A[0]}$ ans += s

A[0] + $\overset{0 — 1}{A[1]}$ ans += s

A[0] + A[1] + $\overset{0 — 2}{A[2]}$ ans += s

L = 1 { A[1] ans += s

A[1] + A[2] ans += s

L = 2 { A[2] ans += s

Carry Forward

```
ans = 0
for L → 0 to (N-1) {
    sum = 0
    for R → L to (N-1) {
        sum += A[R]
        ans += sum
```

```
    }
}
return ans
```

$TC = \underline{O(N^2)}$ $SC = \underline{O(1)}$

→ If one element is used multiple times to calculate the answer → Contribution Technique

$$\boxed{Ans = \sum_{\forall i} \text{contribution of } A[i]}$$

$$A = \begin{array}{cccc} & 0 & 1 & 2 \\ [& 3 & 2 & 5] \end{array}$$

3 → 3

3 2 → 5

3 2 5 → 10

2 → 2

2 5 → 7

5 → 5

$\underline{32}$ (Ans)

$3 * 3 = 9$

$2 * 4 = 8$

$5 * 3 = \underline{15}$

$\underline{32}$ (Ans)

Contribution of A[i]

→ A[i] * (# subarrays A[i] is part of)

$$A = \begin{array}{ccccccc} & 0 & 1 & 2 & 3 & 4 & 5 \\ [& 3 & \textcircled{-2} & 4 & -1 & 2 & 6] \end{array}$$

Ans = $\underline{10}$

L	R
0	1
0	2
0	3
0	4
0	5

L	R
1	1
1	2
1	3
1	4
1	5

$$A = \begin{array}{ccccccc} & 0 & 1 & \overset{i}{2} & 3 & 4 & 5 \\ [& 3 & -2 & \textcircled{4} & -1 & 2 & 6] \end{array}$$

Ans = $\underline{12}$

L	R
0	2
0	3
0	4
0	5

L	R
1	2
1	3
1	4
1	5

L	R
2	2
2	3
2	4
2	5

subarrays with A[i]

\# starting index (L) = [0 i] → $i - 0 + 1 = \underline{i+1}$

\# ending index (R) = [i N-1] → $N-1-i+1 = \underline{N-i}$

\# subarrays having A[i] = $(i+1) * (N-i)$

$$\boxed{Ans = \sum_{\forall i} A[i] * (i+1) * (N-i)}$$

```
ans = 0
for i → 0 to (N-1) {
    ans += A[i] * (i+1) * (N-i)
}  return ans
```

TC = $\underline{O(N)}$ SC = $\underline{O(1)}$

$$A = \left[\begin{array}{ccc} \cancel{i} & \cancel{i} & i \\ 0 & 1 & 2 \\ 3 & 2 & 5 \end{array}\right]$$

$$\begin{aligned} Ans = {} & 3 * 1 * 3 + \\ & 2 * 2 * 2 + \\ & 5 * 3 * 1 = 9 + 8 + 15 = \underline{32} \ ✓ \end{aligned}$$

Q → Find the # of subarrays with length K.

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

N = 6

K = 3 → Ans = $\underline{4}$

K = 5 → Ans = $\underline{2}$

$$\boxed{\begin{array}{l} 1^{st}\text{ subarray} \rightarrow 0 \text{ —— } (K-1) \\ \text{last subarray} \rightarrow (N-K) \text{ —— } (N-1) \end{array}}$$

[st (N-1)]

$N - \cancel{1} - st \cancel{+1} = K \Rightarrow st = \underline{N-K}$

\# values $[(K-1) \_\_\_\_ (N-1)] \rightarrow (N-1)-(K-1)+1 = \underline{N-K+1}$ (Ans)

$N = 7 \quad K = 4 \qquad$ Ans $= N - K + 1 = 7 - 4 + 1 = \underline{4}$

Q → Print all start & end indices of length K.

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

$N = 6 \qquad K = 3$

L	R
0	2
1	3
2	4
3	5

```
for L → 0 to (N-K) {
    R = L + (K-1)
    print (L, R)
}
```

TC = $\underline{O(N)}$ SC = $\underline{O(1)}$

Q → Given an integer array,
find max subarray sum ∀ subarrays
of length K.

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

$N = 6 \qquad K = 3$

5 1 5 7

Ans = $\underline{7}$

L	R
0	2
1	3
2	4
3	5

[L R]

$R - L + 1 = K$

$\Rightarrow \underline{R = K - 1 + L}$

Bruteforce

```
ans = Int_Min
for L → 0 to (N-K) {
    R = L + (K-1)
    sum = 0
    for i → L to R {
        sum + = A[i]
    }                         → TC = O(K)
    ans = max (ans, sum)
}
```

SC = $\underline{O(1)}$

$$TC = O((N-K+1) * K) \rightarrow \underline{O(N^2)}$$

$K = 1$ $(N \cancel{-1} \cancel{+1}) * 1 = N$

$K = N$ $(\cancel{N} - \cancel{N} + 1) * N = N$

$K = \frac{N}{2}$ $(N - \frac{N}{2} + 1) * \frac{N}{2} = (\frac{N}{2} + 1) * \frac{N}{2} = \frac{N^2}{4} + \frac{N}{2} \rightarrow \underline{O(N^2)}$

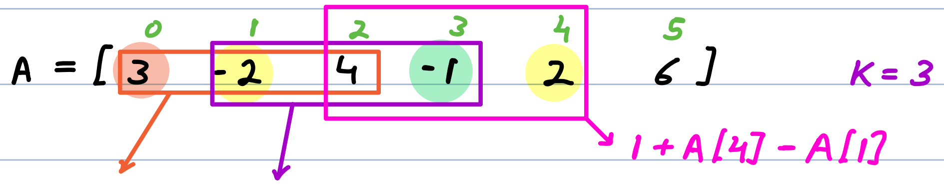
Prefix Sum

```
P[0] = A[0]
for i → 1 to (N-1) {
    P[i] = P[i-1] + A[i]
}

ans = Int_Min
for L → 0 to (N-K) {
    R = L + (K-1)
    if (L == 0) ans = max (ans, P[R])
    else ans = max (ans, P[R] - P[L-1])
}
```

```
return ans
```

TC = O(N + (N-K+1)) → O(N)

SC = O(N)



5

5 + A[3] - A[0]

= 1

1 + A[4] - A[1]

= 5

Sliding Window → Fixed length subarray

L	R	sum
0	2	A[0] + A[1] + A[2] = 5
1	3	sum + A[3] - A[0] = 1
2	4	sum + A[4] - A[1] = 5
3	5	sum + A[5] - A[2] = 7

max → 7 (Ans)

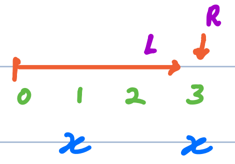
```
sum = 0
for i → 0 to (K-1) {
        sum += A[i]
}

ans = sum
for R → K to (N-1) {
        L1 = R - K
        sum += A[R] - A[L1]
        ans = max(ans, sum)
}
return ans
```

R - L + 1 = K ⇒ L = R - K + 1

L —— R

TC = O(K + N - K) = O(N)

SC = O(1)



$$A = \left[\begin{array}{cccccc} 0 & 1 & 2 & 3\checkmark & 4\checkmark & 5\checkmark \\ 3 & 2 & 4 & 2 & 3 & 4 \end{array}\right]$$

L R

vst[3] = true

$$Ans = \sum_{\forall R} \#\text{ subarrays ending at } R$$

$1 + 2 + 3 + 2 + 3 + 3 = \underline{14}$

$\underline{R - L + 1}$
