



Introduction to Arrays

$A = [1 \ 2 \ 3 \ 4 \ \dots \ n]$
 0 1 2 3 n-1

Indexing starts from 0

< Question > : Print all elements of the array

$A = [1, 2, 3]$

TC: $O(N)$

print(A)

▼ Print Array

for i in A:
 print(i)

TC: to access $A[i]$

TC: $O(1)$

[6]
 ✓ Os

```
▶ A = [5, 2, 9, 6, 1]
# Option 1
print(A)

print("#####")
# Option 2
for i in range(len(A)):
    print(A[i], end = " ")

print("#####")
# Option 3
for val in A:
    print(val, end = " ")
```

```
⇒ [5, 2, 9, 6, 1]
#####
5 2 9 6 1 #####
5 2 9 6 1
```

```
int arr[5] = {5,-4,8,9,10};
```

A = [5 , -4 , 8 , 9 , 10]
 0 1 2 3 4

Find sum of 1st and 5th element

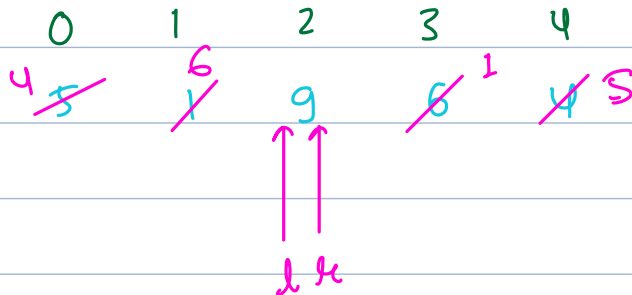
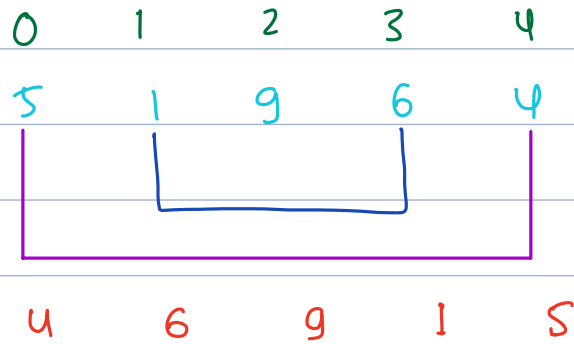
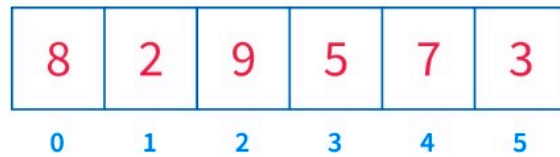
```
print ( A[0] + A[4] )
```



2. Reverse the given array



Reverse(arr)





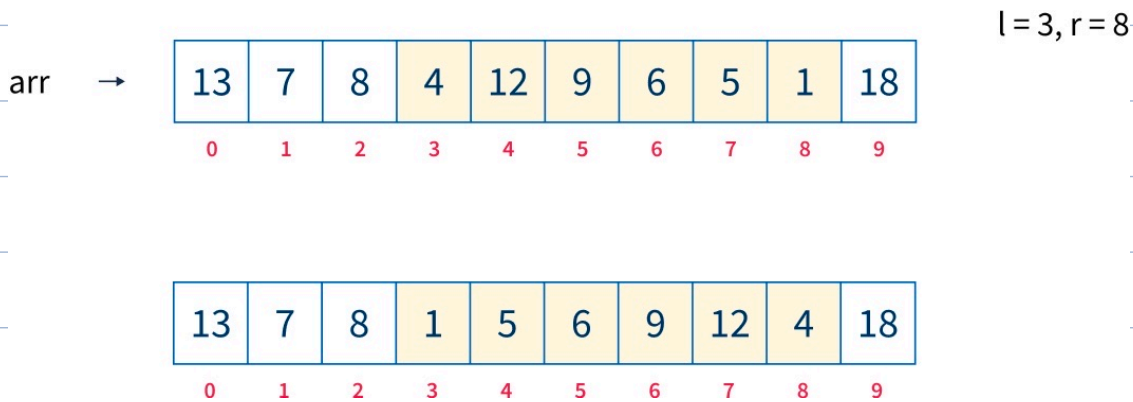
```
A = [5, 9, 2, 1, 6, 8]
#   l               r
print("before", A)
l = 0
r = len(A) - 1

while l < r:
    A[l], A[r] = A[r], A[l] # swap
    #               8, 5
    print(A)
    l += 1
    r -= 1

print("after", A)
# TC : O(N)
# SC : O(1)
```



3. Reverse part of an array



$A = [5, 9, 6, 2, 4, 1]$

Reverse a part of array from index s to index e

$A = [5, 9, 6, 2, 4, 1]$

$s = 2$
 $e = 4$

def reverse(A, s, e):

$l = s$

$r = e$

 while $l < r$:

$A[l], A[r] = A[r], A[l]$

$l += 1$

$r -= 1$

T.C: $O(e-s)$

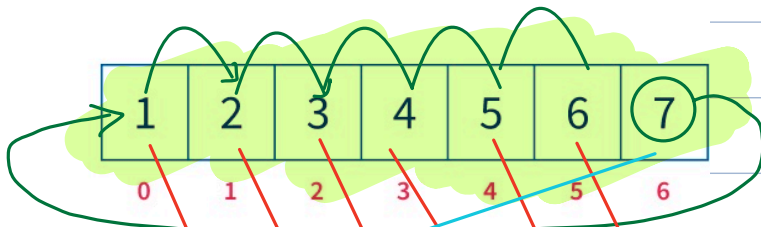
↓
 $O(N)$

**Rotate an array K times**

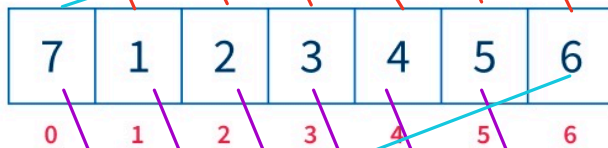
→ No extra space allowed

SC : $O(1)$

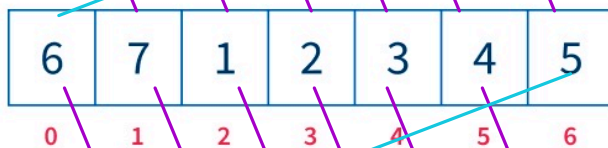
N=7



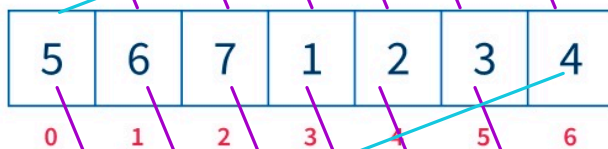
K=1



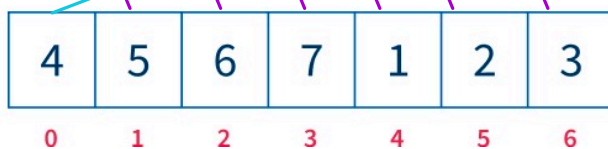
K=2



K=3



K=4




```
A = [5, 9, 2, 1, 6, 8]
```

```
def rotate(A): # Rotating the array one step towards the right
    temp = A[-1] # store the last value
```

```
        # [start, end)
```

```
    for i in range(len(A) - 1, 0, -1): # N-1....1
```

```
        A[i] = A[i-1]
```

```
    A[0] = temp
```

```
print("before", A)
```

```
K = 3
```

```
for _ in range(K): # Loop runs exactly K times
```

```
    rotate(A)
```

```
print("after", A)
```

```
#####
```

```
# TC:  $O(KN)$ 
```

```
# SC:  $O(1)$ 
```




Optimisation

$N=7$

1	2	3	4	5	6	7
0	1	2	3	4	5	6

$k=3$

$A = [1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7]$ I/p

$reverse(A, 0, N-1)$ 7 6 5 4 3 2 1

$reverse(A, 0, k-1)$ 5 6 7 4 3 2 1

$reverse(A, k, N-1)$ 5 6 7 1 2 3 4

$[5 \ 6 \ 7 \ 1 \ 2 \ 3 \ 4]$ O/p

Idea \longrightarrow

$reverse(A, 0, N-1)$

$reverse(A, 0, k-1)$

$reverse(A, k, N-1)$

$k=100$

$N=7$

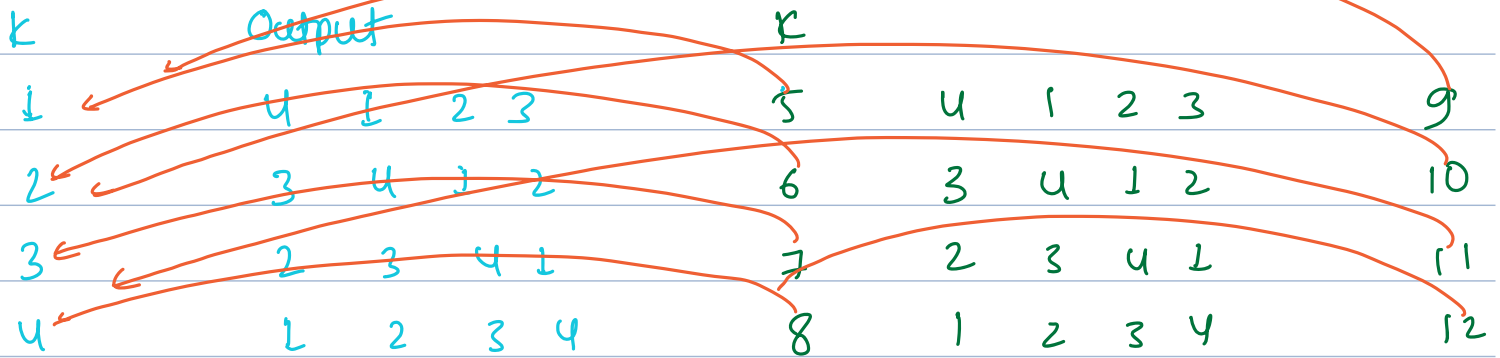
$A = [1, 2, 3, 4, 5, 6, 7]$

Note : Our idea doesn't work for $k > N$

if $k > N$:

$$k = k \% N$$

$A = [1, 2, 3, 4]$



def rotate k times (A , k) :

TC: $O(N)$

$N = \text{len}(A)$

$k = k \% N$

reverse (A , 0 , $N-1$)

reverse (A , 0 , $k-1$)

reverse (A , k , $N-1$)

N
 k
 $N-1-k+1$
 $N-k$

$2N$

TC: $O(N)$

Break : 9 : 46

A = []

A.append(1) # [1]

A.append(2) # [1, 2]

A.pop() # [1]

Google stock

changes = [-5 , 10 , 20 , 40 , 50 , -10 , 80 , -90 , -20 , -10]
 0 1 2 3 4 5 6 7 8 9

Net change in the stock price from day 0 to day 9

Start day	End day	change
0	9	65
1	4	120
0	0	-5
7	9	-120
4	6	120



< Question > : Given an array of N integers and Q queries. For each query calculate the sum of elements in the range - [L , R]

Note : L and R are indices such that $L \leq R$.

$1 \leq N, Q \leq 10^5$

arr[10] → [-3 6 2 4 5 2 8 -9 3 1]

0 1 2 3 4 5 6 7 8 9

Queries

L R

4

8

→ 9

3

7

→ 10

1

3

→ 12

0

4

→ 14

7

7

→ -9

Bruteforce → For every query
print sum from L...R

```
#      0    1    2    3    4
A = [1, 2, -1, 4, 9]
Q = [
    [0, 1], # Q1
    [3, 4]  # Q2
]
```

TC: $O(Q*N)$

SC: $O(1)$

```
print(A)
for L, R in Q:  $\longrightarrow$  Q times
    total = 0
    for i in range(L, R+1):  $\longrightarrow$  N times
        total += A[i]
    print([L, R], "total = ", total)
```

```
[1, 2, -1, 4, 9]
[0, 1] total = 3
[3, 4] total = 13
```



- Given Royal Challengers Bengaluru's cricket scores for first 10 overs of batting.

	OVERS	1	2	3	4	5	6	7	8	9	10
	SCORE	2	8	14	29	31	49	65	79	88	97

- Runs scored in 7th over = $A[7] - A[6]$

- Runs scored from 6 - 10th over = $\begin{aligned} &\text{run after 10} = A[10] \\ &- \text{run after 5} = A[5] \end{aligned} = A_{10} - A_5 = 97 - 31 = 66$

- Runs scored in 10th over = $A_{10} - A_9 = 97 - 88 = 9$

	OVERS	1	2	3	4	5	6	7	8	9	10
	SCORE	2	8	14	29	31	49	65	79	88	97

- Runs scored from 3 - 6th over = $A_6 - A_2 = 49 - 8 = 41$

- Runs scored from 4 - 9th over = $A_9 - A_3 = 88 - 14 = 74$

- Runs scored in lth - rth over = $A[l] - A[l-1]$
if $l \neq 0$



Prefix sum = Cumulative sum.

How to create psum()

arr[10] → [-3 6 2 4 5 2 8 -9 3 1]

0 1 2 3 4 5 6 7 8 9

psum[10] = [-3 3 5 9 14 16 24 15 8 19]

arr[6] → [10 32 6 12 20 1]

psum[10] = [10 42 48 60 80 81]

$$p = [0] * N$$

$$p[0] = A[0]$$

for i in range(1, N):

$$p[i] = p[i-1] + A[i]$$

$$\Rightarrow p = \text{list}(\text{itertools.accumulate}(A))$$



arr[10] → ^{0 1 2 3 4 5 6 7 8 9}
[-3 6 2 4 5 2 8 -9 3 1]

^{0 1 2 3 4 5 6 7 8 9}
P() -3 3 5 9 14 16 24 15 18 19

Queries

L R

3 4 $P[4] - P[2] = 14 - 5 = \underline{\underline{9}}$

0 5 $P[5] = 16$

6 7 $P[7] - P[5] = -1$

[21]

▶ # Optimisation using prefix sum

0 1 2 3 4

A = [1, 2, -1, 4, 9]

Q = [
[0, 1], # Q1
[3, 4] # Q2
]

import itertools

print(A)

P = list(itertools.accumulate(A)) # This calculate prefix sum

print(P)

for L, R in Q:

if L == 0:

total = P[R]

else:

total = P[R] - P[L-1]

print([L, R], "total = ", total)

⇒ [1, 2, -1, 4, 9]
[1, 3, 2, 6, 15]
[0, 1] total = 3
[3, 4] total = 13

} TC: O(N)

} TC: O(Q)

TC: O(N+Q)

SC: O(N)

prefix sum

O(1)

By using same array for Psum



Modification of same array into psum

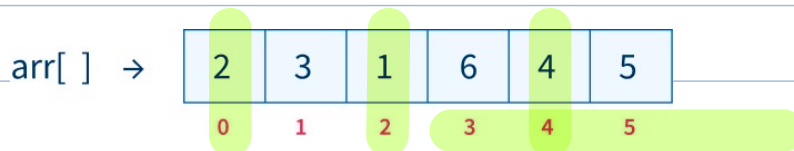
$$A = [1, \overset{3}{\cancel{2}}, \overset{6}{\cancel{3}}]$$
$$p =$$

Goal \longrightarrow Instead of creating addⁿ array
use given array itself.

for i in range $(1, N-1)$:
 $A[i] = A[i] + A[i-1]$

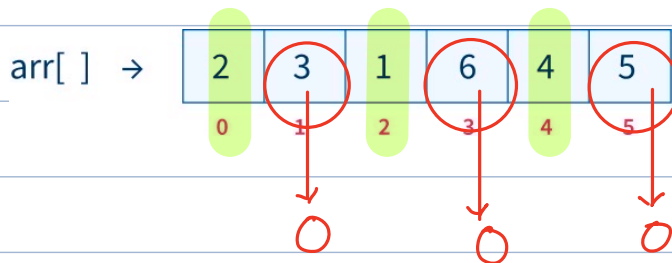


< **Question** > : Given an arr[N] and Q queries with start(s) and end(e) index. For every query print sum of all even indexed elements from s to e.



Queries

s	e	
1	3	→ 1
2	5	→ 5
0	4	→ 7
3	3	→ 0



∴ No point of odd index value change them to 0

0	1	2	3	4	5	} prev Q.
2	0	1	0	4	0	

TC: $O(N+Q)$
 SC: $O(1)$

$$A = \begin{bmatrix} 2 & \cancel{0} & 3 & \cancel{0} & 5 \\ 0 & 1 & 2 & 3 & 4 \end{bmatrix}$$

