Rotate Array By k

Ex :- arr[] = (3, -2, 1, 4, 6, 9, 8) , k = 3

We have to rotate the array by k (3) places

let's observe initial input and final output once,

Input :- (3, -2, 1, 4, 6, 9, 8)

Output :- (6, 9, 8, 3, -2, 1, 4)

Observations

① if k = 3 → input's last 3 elements came to Starting & became first 3 Elements

(Sequence also not changed)

Input :- (3, -2, 1, 4, 6, 9, 8)

Output :- (6, 9, 8, 3, -2, 1, 4)

Observations

① if k = 3 ↴
leave last 3 Elements and see...,

those elements occured after k elements in the output.

☆ In reverse Order

* So... figure out a ways to bring last k Elements to first and first (n-k) Elements to back (as reversed)

↓

Our answer array is Set

Algorithm :

① Reverse whole array

② Reverse first k-elements

③ Reverse the elements, which are after k elements

Ex: arr[] = (3, -2, 1, 4, 6, 9, 8) , k = 3

⇓

step 1: (8, 9, 6, 4, 1, -2, 3) ① Reverse whole array

⇓

Step 2: (6, 9, 8, 4, 1, -2, 3) ② Reverse first k-elements

⇓

Step 3: (6, 9, 8, 3, -2, 1, 4) ③ Reverse the elements, which are after k elements

This is our final array

* Create a reverse function
  - which takes array and (L, R) → Ranges
    - from where to where array need to be reversed

* Call the reverse function for 3 times
  1. reverse (arr, 0, n-1) → whole array reversing
  2. reverse (arr, 0, k-1) → first k-Elements
  3. reverse (arr, k, n-1) → last Elements from k