1.Easy\06.Rotate_array_left&right_by_k_places.cpp

```
/*
QUESTION:-

Given an integer array nums, rotate the array to the right by k steps, where k is non-negative.

Example 1:

Input: nums = [1,2,3,4,5,6,7], k = 3
Output: [5,6,7,1,2,3,4]
Explanation:
rotate 1 steps to the right: [7,1,2,3,4,5,6]
rotate 2 steps to the right: [6,7,1,2,3,4,5]
rotate 3 steps to the right: [5,6,7,1,2,3,4]
Example 2:

Input: nums = [-1,-100,3,99], k = 2
Output: [3,99,-1,-100]
Explanation:
rotate 1 steps to the right: [99,-1,-100,3]
rotate 2 steps to the right: [3,99,-1,-100]
*/

/*
APPROACH:-
To rotate the array k places to right follow below steps
-> Reverse first n-k elements
-> Reverse last k elements
-> Reverse the entire array

To rotate the array k places to left follow below steps
-> Reverse first k elements
-> Reverse last n-k elements
-> Reverse the entire array
*/

// CODE:-

// RIGHT ROATATE:-
void rightRotate(int arr[], int n, int k)
{
    k = k % n; // to keep k within the range
    reverse(arr, arr + (n - k));
    reverse(arr + (n - k), arr + n);
    reverse(arr, arr + n);
}

// LEFT ROATATE:-
void leftRotate(int arr[], int n, int k)
{
    k = k % n; // to keep k within the range
    reverse(arr, arr + k);
```

```
    reverse(arr + k, arr + n);
    reverse(arr, arr + n);
}

// TIME COMPLEXITY = O(N)
// SPACE COMPLEXITY = O(0)
```