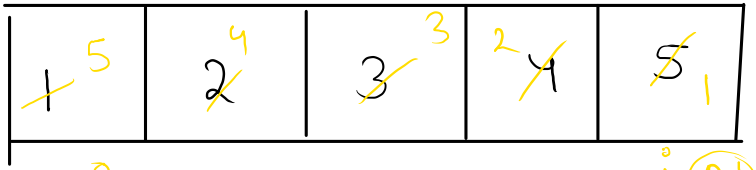
Reverse.

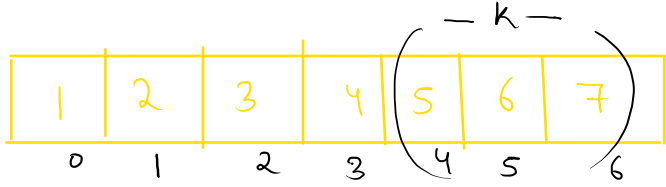
inplace solution. w/o taking extra space.




```
while( i < j)
{
    swap(i, j)
    i++
    j--
}
```

. Rotate Array

```
Input: nums = [1,2,3,4,5,6,7], k = 3
Output: [5,6,7,1,2,3,4]
```



k = 8

n = 7

$0 \le k \le 10^5$

k = 14

k = 15

k = 0 1 2 3 4 5 6

k = 1 7 1 2 3 4 5 6

k = 2 6 7 1 2 3 4 5

k = 3 5 6 7 1 2 3 4

k = 4 4 5 6 7 1 2 3

k = 5 3 4 5 6 7 1 2

k = 6 2 3 4 5 6 7 1

k = 7

k = 8

← k = 9

1 2 3 4 5 6 7

7 1 2 3 4 5 6

k = k % n

k = 8 → k % n

8 % 7

= 1

k = 9 → 9 % 7

= 2

k = 5

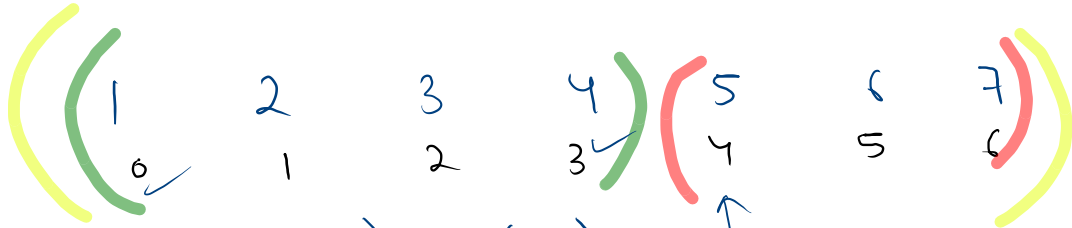
5 % 7

= 5

k = 2.

6 7 1 2 3 4 5

k = 3



5 6 7 1 2 3 4 (0, 7-3-1) ⇒ (0, 3)

n-k = 7-3 = {4, n-1} (0, n-k-1) (n-k, n-1) (0, n-1)

5 6 7

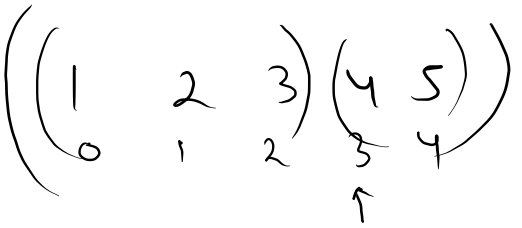
order.
Green.
Red
Yellow.

4 3 2 1

4 3 2 1 7 6 5

5 6 7 1 2 3 4

k=2
((1 2 3) (4 5))
4 5 1 2 3
l, r
3 2 1 5 4
4 5 1 2 3



$k = 2$

$n = 5$

$n - k - 1$

Rotate.

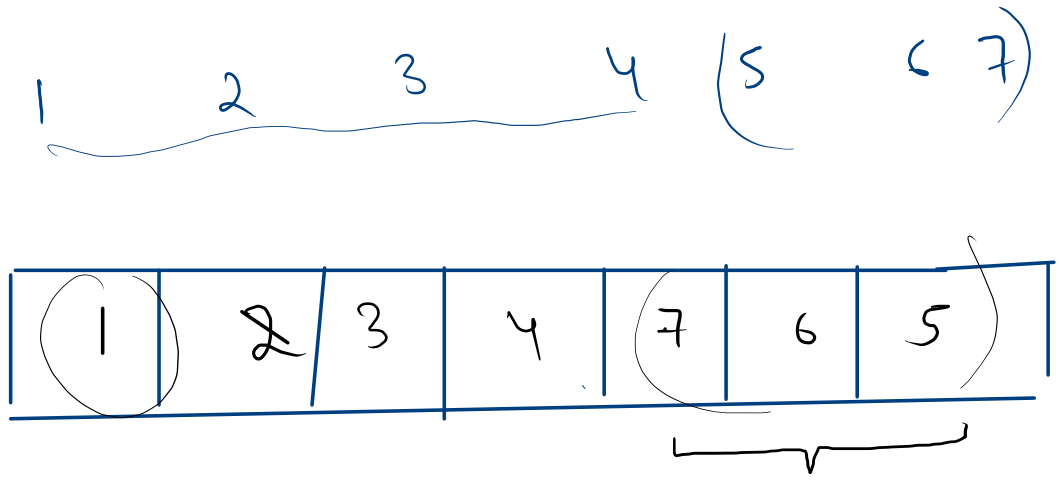
'k'

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

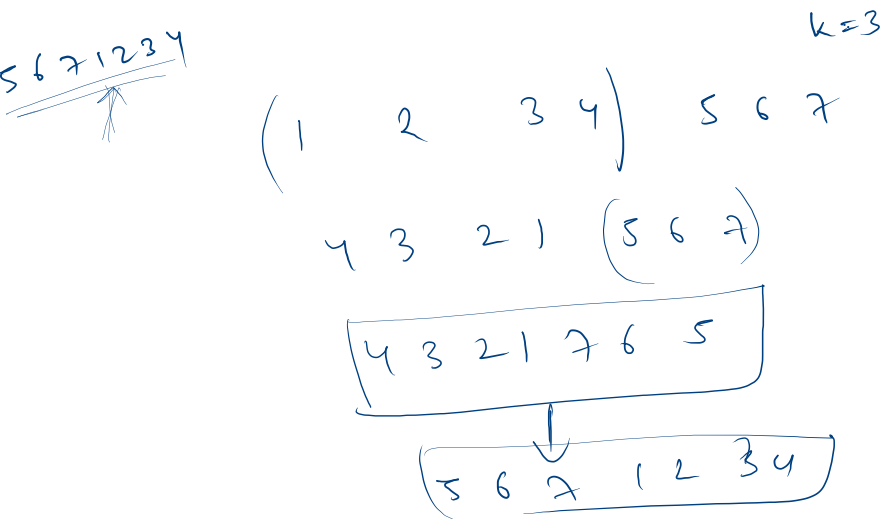
reverse (A , $n-k$, $n-1$)

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

K=3



```java
public void reverse(int [] A, int left, int right){
    int i = left;
    int j = right;

    while(i < j){
        int tmp = A[i];
        A[i] = A[j];
        A[j] = tmp;

        i++;
        j--;
    }
}

public void rotate(int[] nums, int k) {
    int n = nums.length;
    k = k % n;      // k is in the range of n

    reverse(nums, 0, n-k-1 );
    reverse(nums, n-k, n-1 );
    reverse(nums, 0, n-1);
}
```



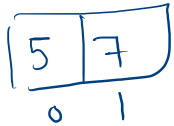
$17 \% 5 \longrightarrow$ ans ≥ 5

$17 \% 3 \longrightarrow$ ans ≥ 3

$k \% n \longrightarrow$ ans $\geq n$

n = 2

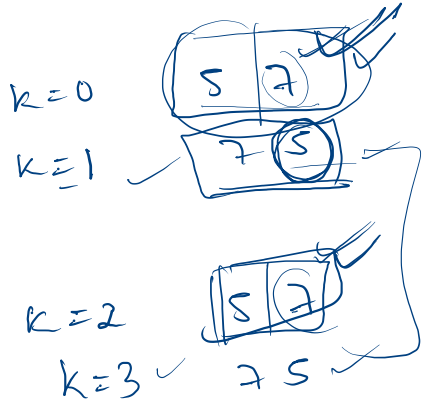
k = 3



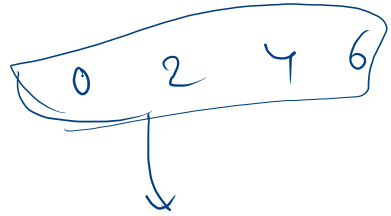
k = k % n

3 % 2 = 1

k = 1



k = 2



4 % 2 = 0

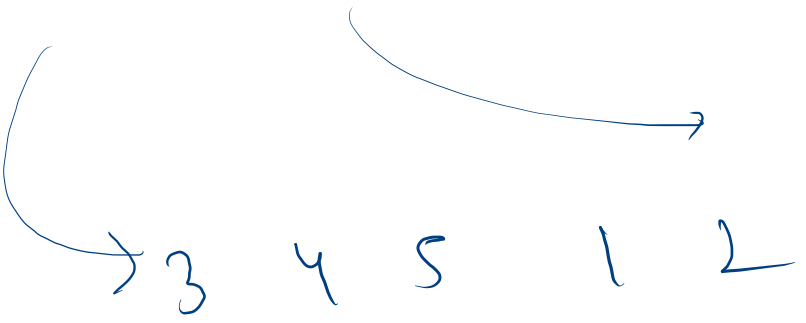
k = = 0

Follow up

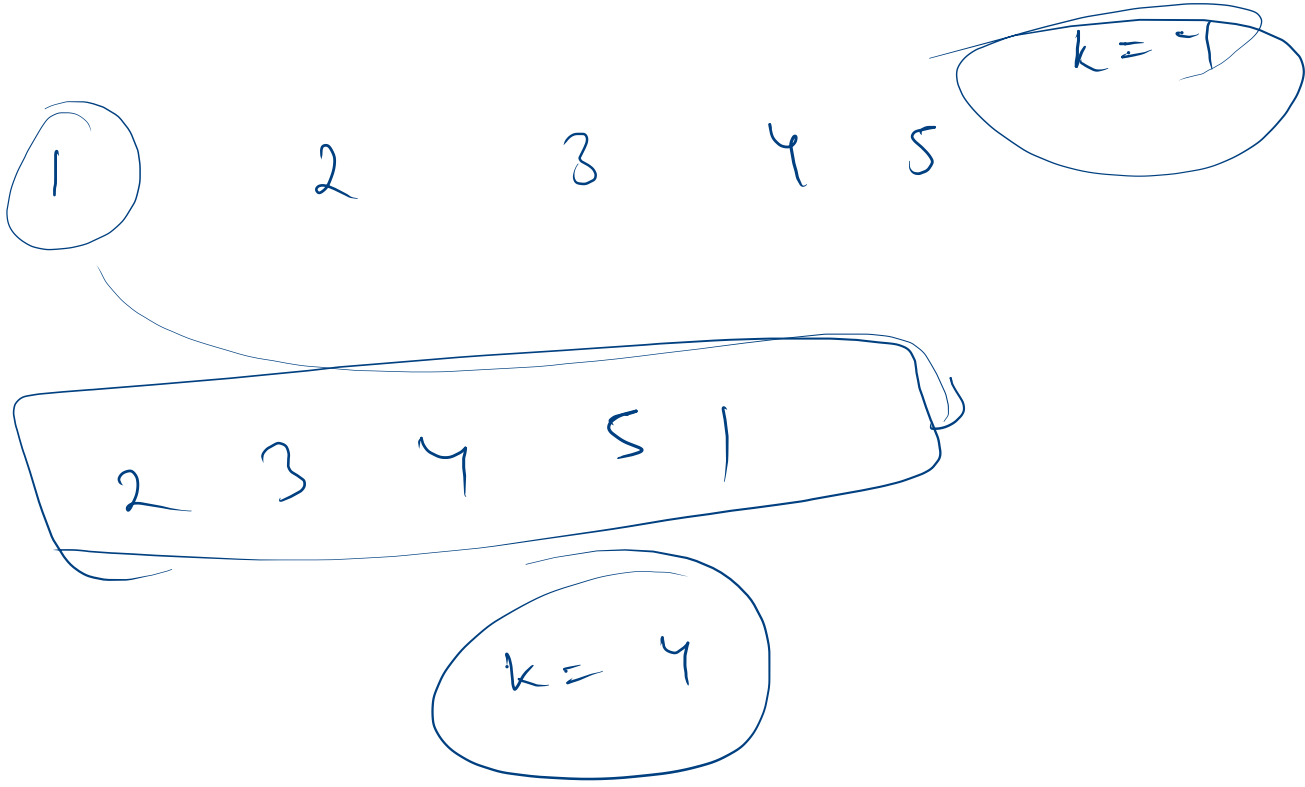
Try by yourself.

K = 2

1 2 3 4 5



HW



$k = 2$

eg. $(1\ 2\ 3\ 4\ 5)(6\ 7)$

$5\ 4\ 3\ 2\ 1\ 7\ 6$

$\downarrow$

$6\ 7\ 1\ 2\ 3\ 4\ 5$

1470. Shuffle the Array

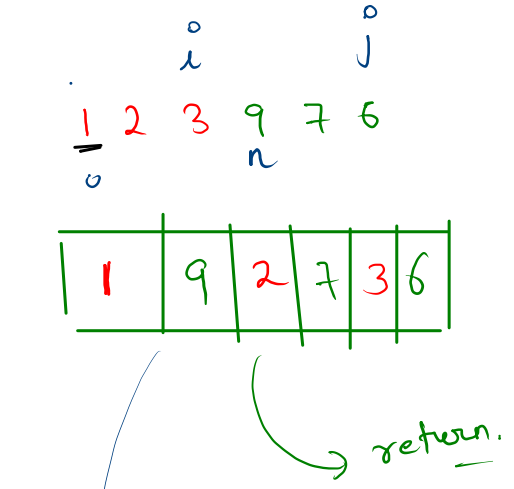
Easy 4988 255 Add to List Share

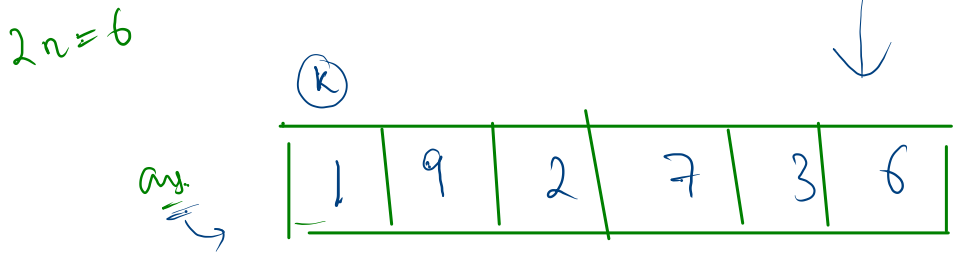
Given the array `nums` consisting of `2n` elements in the form `[x1,x2,...,xn,y1,y2,...,yn]`.

Return the array in the form `[x1,y1,x2,y2,...,xn,yn]`.

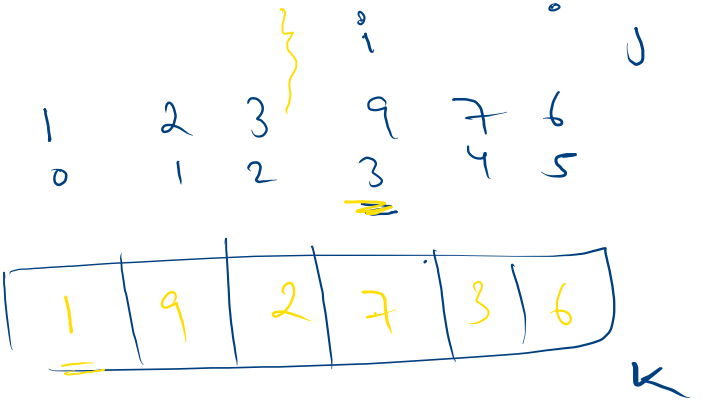
Example 1:

```
Input: nums = [2,5,1,3,4,7], n = 3
Output: [2,3,5,4,1,7]
Explanation: Since x1=2, x2=5, x3=1, y1=3, y2=4, y3=7 then the answer is
[2,3,5,4,1,7].
```



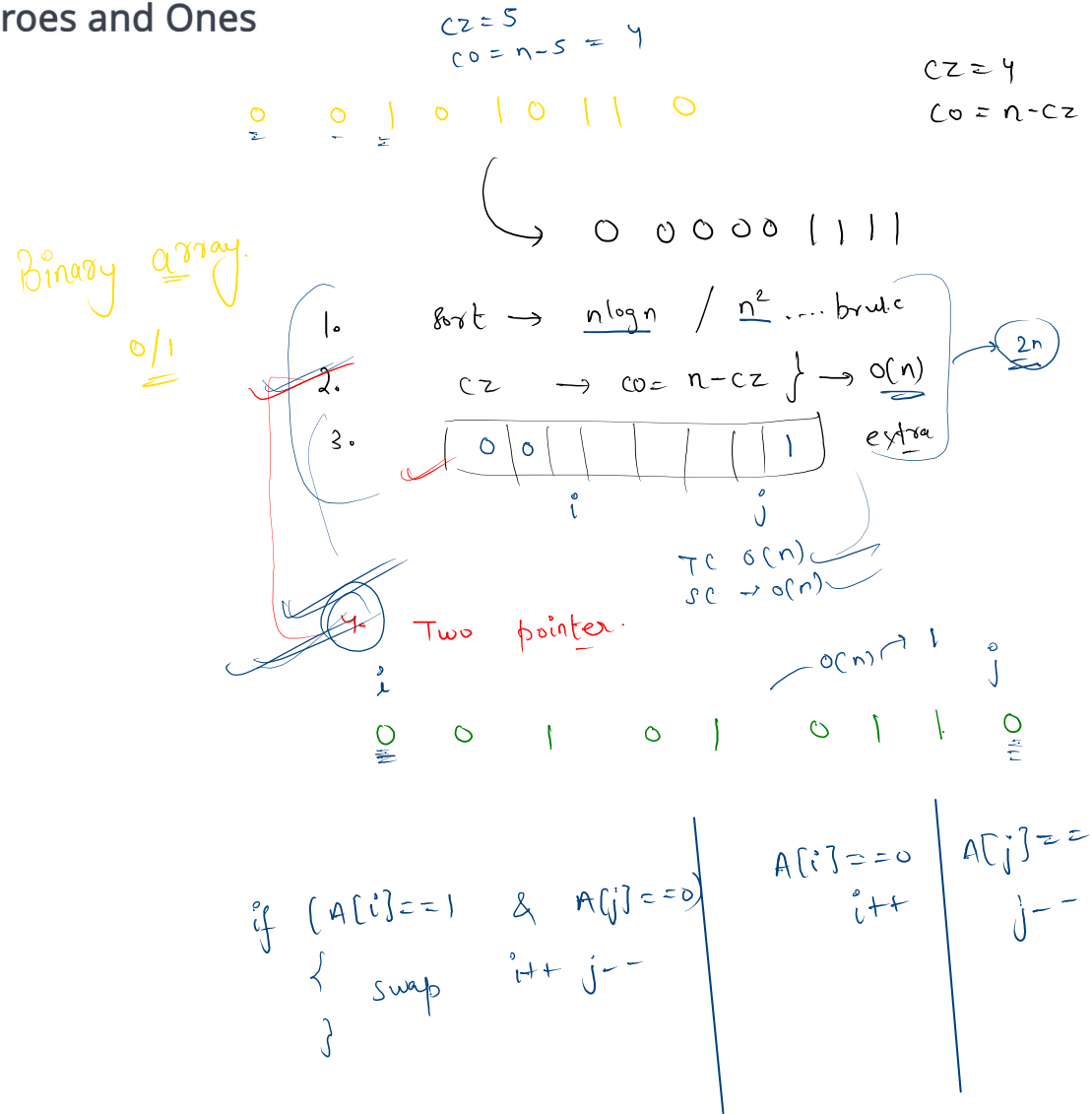


n=3



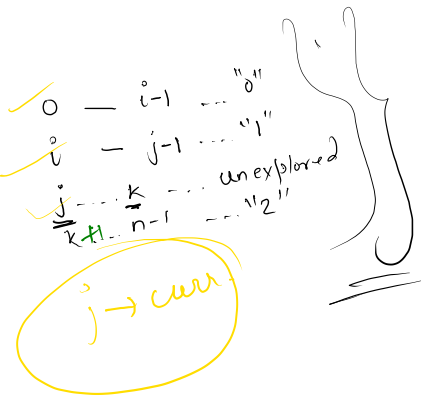

```java
class Solution {
    public int[] shuffle(int[] nums, int n) {
        int [] ans = new int[nums.length];

        // int [] ans = new int[n * 2];

        int i = 0;
        int j = n;
        int k = 0;

        while(i < n){
            ans[k] = nums[i];
            k++;
            i++;
            ans[k] = nums[j];
            k++;
            j++;
        }

        return ans;
    }
}
```

Zeroes and Ones



```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int [] A = new int [n];

    for (int i = 0; i < n; i++){
        A[i] = scn.nextInt();
    }

    int i = 0;
    int j = n-1;
    while(i < j){
        if(A[i] == 1 && A[j] == 0){
            int tmp = A[i];
            A[i] = A[j];
            A[j] = tmp;
            i++;
            j--;
        }
        else if(A[i] == 0){
            i++;
        }
        else if(A[j] == 1){
            j--;
        }
    }

    for( i = 0; i < n; i++){
        System.out.print(A[i] + " ");
    }

}
```

Dutch National Flag.

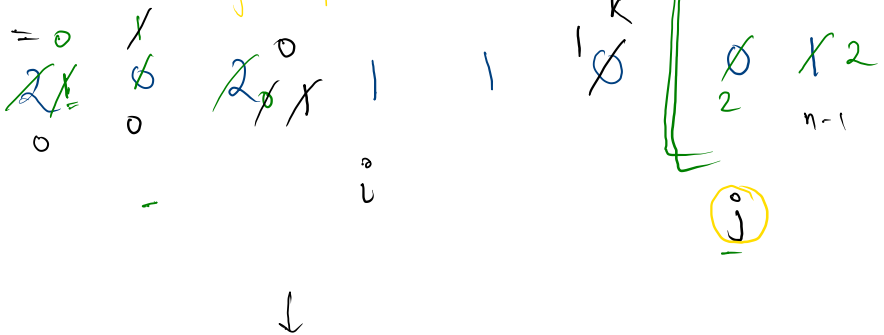


A[j] == 0
swap(i, j)
i++ j++

A[j] == 1
j++

A[j] == 2
swap(j, k)
k--

j ≤ k
Work



↓

0 0 0 1 1 1 2 2