Zeroes and Ones

Sample Input 0

```
6
0 1 1 1 1 0
```

Sample Output 0

```
0 0 1 1 1 1
```

i

j

if A[i] == 0
i++

else if A[j] == 1
j--

else swap
i++
j--

i<j

```java
import java.io.*;
import java.util.*;

public class Solution {

    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, j = n-1;
        //logic
        while(i < j){
            if(A[i] == 0){
                i++;
            }
            else if(A[j] == 1){
                j--;
            }else{
                int tmp = A[i];
                A[i] = A[j];
                A[j] = tmp;
                i++;
                j--;
            }
        }
        //print
```

Rotate Right

Problem | Submissions | Leaderboard | Discussions

John is an athlete who has been training for a marathon for months. He uses a fitness app to track his progress, which stores his daily running data in an array. However, due to a bug in the app, the data is stored is not in order. To fix this, John needs to **rotate** the array to the **right** by a certain number of steps.

Can you help John by writing a program that rotates the array to the right by **k** steps, where **k** is a **non-negative** integer?

Sample Input 0

```
7
1 2 3 4 5 6 7
2
```

Sample Output 0

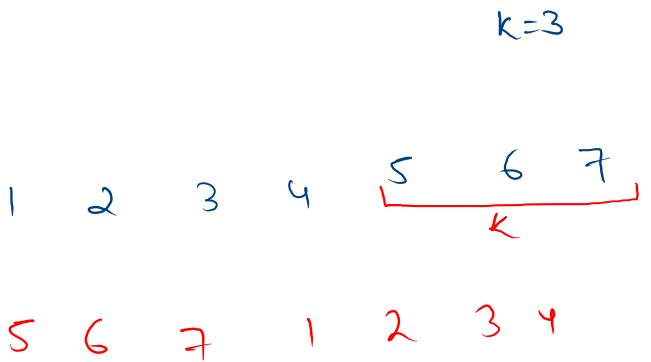
```
6 7 1 2 3 4 5
```

189. Rotate Array

Medium 17425 1912 Add to List Share

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

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



n=7

	1	2	3	4	5	6	7
k=0							
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	1	2	3	4	5	6	7
k=8	7	1	2	3	4	5	6
k=9	6	7	1	2	3	4	5

k=0
k=1

k = 8 k%n

8 % 7 = 1

k = 9

9 % 7 = 2

n=7

k = 16

[0, 6]

k % n

[0, n-1]

16 % 7 = 2

k = 9

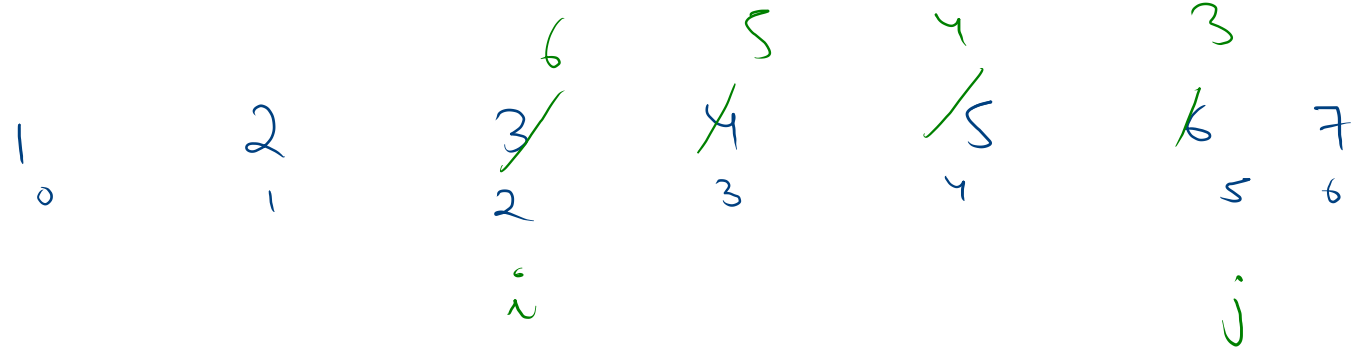
9 % 7 = 2

$K = 5$

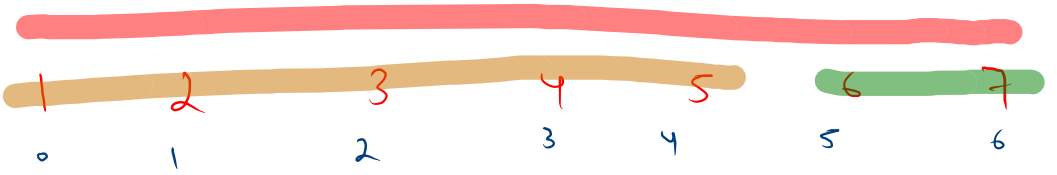
$n = 7$

$K = K \% n$

$K = ?$ (5)



reverse in range $[2, 5]$



5 4 3 2 1 7 6

6 7 1 2 3 4 5

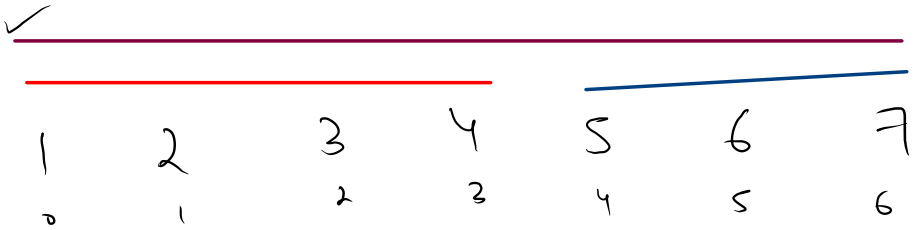
brown

green

red

5 6 7 1 2 3 4

k = 3



4 3 2 1 7 6 5

5 6 7 1 2 3 4

reverse (A, 0, ?)

reverse (A, 1, n-1)

reverse (A, 0, n-1)

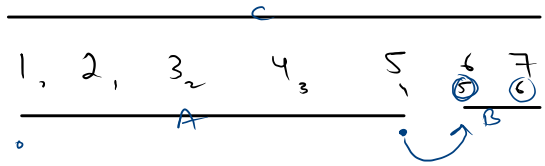
k = 2

6 7 1 2 3 4 5

```java
class Solution {
    public void reverse(int [] A, int i, int j){
        //reverse in range
        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;
        reverse(nums, 0, n-k-1);
        reverse(nums, n-k, n-1);
        reverse(nums, 0, n-1);

    }
}
```



7 - 2 = 5

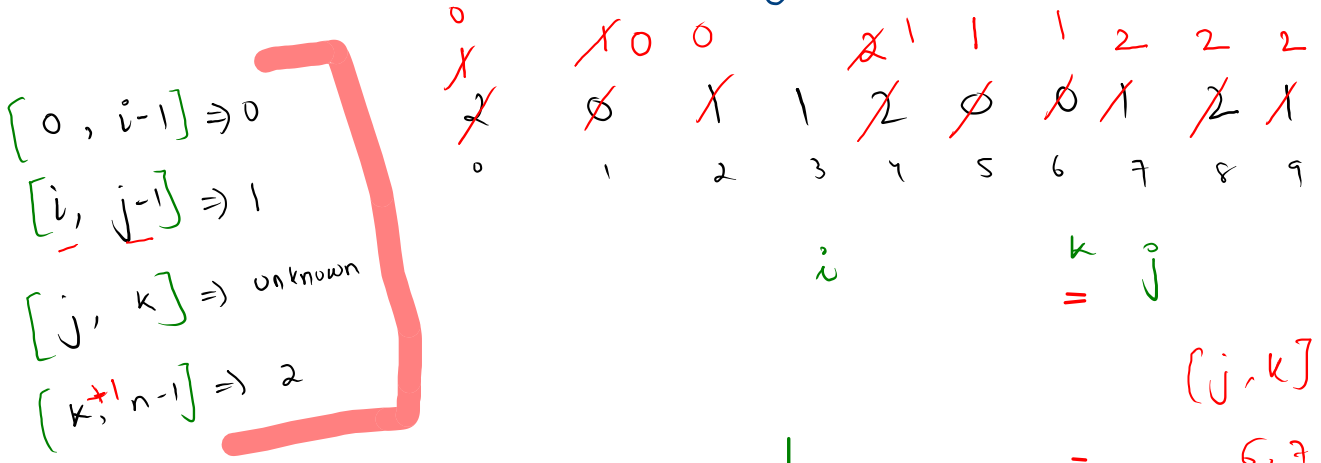
0, n-k-1

0, 7-2-1

n-k +1 +1

n-k

Sort 0 1 2 → Dutch National Flag Algo.



```java

    public void swap(int [] A, int i, int j){
        int tmp = A[i];
        A[i] = A[j];
        A[j] = tmp;
    }

    public void sortColors(int[] nums) {
        int i = 0;
        int j = 0;
        int k = nums.length-1;

        while(j <= k){
            if(nums[j] == 0){
                swap(nums, i, j);
                i++;
                j++;
            }else if(nums[j] == 1){
                j++;
            }else{
                swap(nums, j, k);
                k--;
            }
        }

```

Reach Target

Problem | Submissions | Leaderboard | Discussions

Take the target as an integer input. Then print the **indices** of the two numbers such that they add to the **target**. Note that the array is sorted here.

Use **Two pionter**, answer must be **unique**.

tar = 4

Sample Input 0

```
6
-1 1 2 3 4 5
4
```

tar = 3

o/p
0,4
1,2

-1	1	2	3	4	5
0	1	2	3	4	5
	i				
		j			

Sample Output 0

```
0 5
1 3
```

sum = ~~4~~ ~~3~~ 4
3

sum > tar
j--

sum == tar
print(i,j)
++ i
--j

sum < tar
i++

tar = 3

i < j

-1₀ 1₁ 2₂ 3₃ 4₄ 5₅

j i

sum = 4 3