

LC-189 Reverse Array

	1	2	3	4	5	6	7		
k=0	↔							7	14
k=1	7	1	2	3	4	5	6	8	15
k=2	6	7	1	2	3	4	5	9	16
k=3	5	6	7	1	2	3	4	10	17
k=4	4	5	6	7	1	2	3	11	18
k=5	3	4	5	6	7	1	2	12	19
k=6	2	3	4	5	6	7	1	13	20
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		

$n=7 \rightarrow 7$ unique not

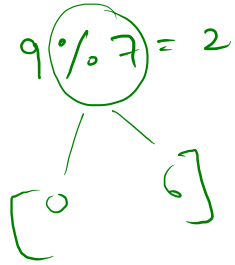
$n=5 \rightarrow 5$ unique not

$n = 7$

$k \% n$

k ↓

0	7
1	8
2	↔ 9
3	↔ 10
4	↔ 11
5	
6	
·	



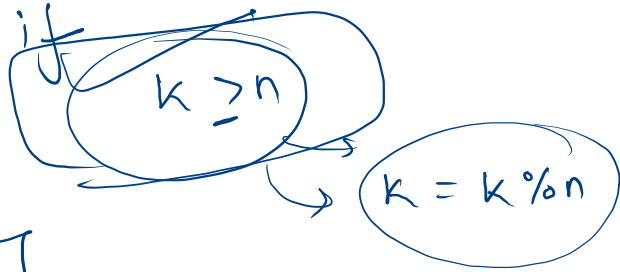
$n = 5$

$k = 18$

k

0	5	10	15
1	6	11	16
2	7	12	17
3	8	13	18
4	9	14	

$18 \% 5 = 3$



eg.

$n = 7$

$k = 4$



$k = k \% n$

$k = 4 \% 7 = ? = 4$

30 40 50 60 70 10 20



$n=7$

$k=5$

$n-k-1$

20 10 70 60 50 40 30

```

1 class Solution {
2
3     public void reverse(int [] A, int i, int j){
4         while(i < j){
5             int tmp = A[i];
6             A[i] = A[j];
7             A[j] = tmp;
8             i++;
9             j--;
10        }
11    }
12
13    public void rotate(int[] nums, int k) {
14        int n = nums.length;
15
16        k = k % n;
17
18        reverse(nums, 0, n-k-1);
19        reverse(nums, n-k, n-1);
20        reverse(nums, 0, n-1);
21    }
22 }

```

30 40 50 60 70 10 20

1 2 3 4 5 6 7

$k=3$

4

5

6

Rotate by left. (HW).

$k=3$

$k = -ve.$

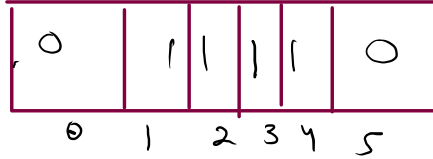
10 20 30 40 50 60 70

RBL 40 50 60 70 10 20 30

RBR 50 60 70 10 20 30 40

Sort 0 1

$n = 6$



Count Z = 2

Count one = $n - 2 = 4$

Sample Input 0

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

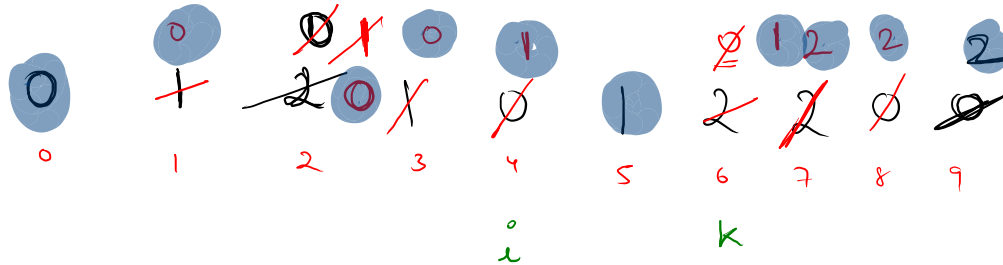
Sample Output 0

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

sort \rightarrow B } n^2
 \rightarrow I
 \rightarrow S
 \rightarrow Arrays.sort
 \rightarrow $O(n \log n)$
 \rightarrow $O(n)$.
 \downarrow
2 pointers
 \downarrow
Sort by parity.

```
2 import java.util.*;
3
4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner scn = new Scanner(System.in);
8         int n = scn.nextInt();
9         int [] A = new int[n];
10        for(int i = 0; i < n; i++){
11            A[i] = scn.nextInt();
12        }
13
14        int i = 0;
15        int j = n-1;
16        while(i < j){
17            if(A[i] == 0){
18                i++;
19            }else if(A[j] == 1){
20                j--;
21            }else{
22                int tmp = A[i];
23                A[i] = A[j];
24                A[j] = tmp;
25                i++;
26                j--;
27            }
28        }
29
30        for(i = 0; i < n; i++){
31            System.out.print(A[i] + " ");
32        }
33    }
34 }
```

Sort 0 1 2 / Dutch National Flag Algo / Sort Colors / LC 75



$j \rightarrow \text{move}$

$0 \rightarrow i-1 = 0$

$i \rightarrow j-1 = 1$

$j \rightarrow k = \text{unexplored}$

$k+1 \rightarrow n-1 = 2$

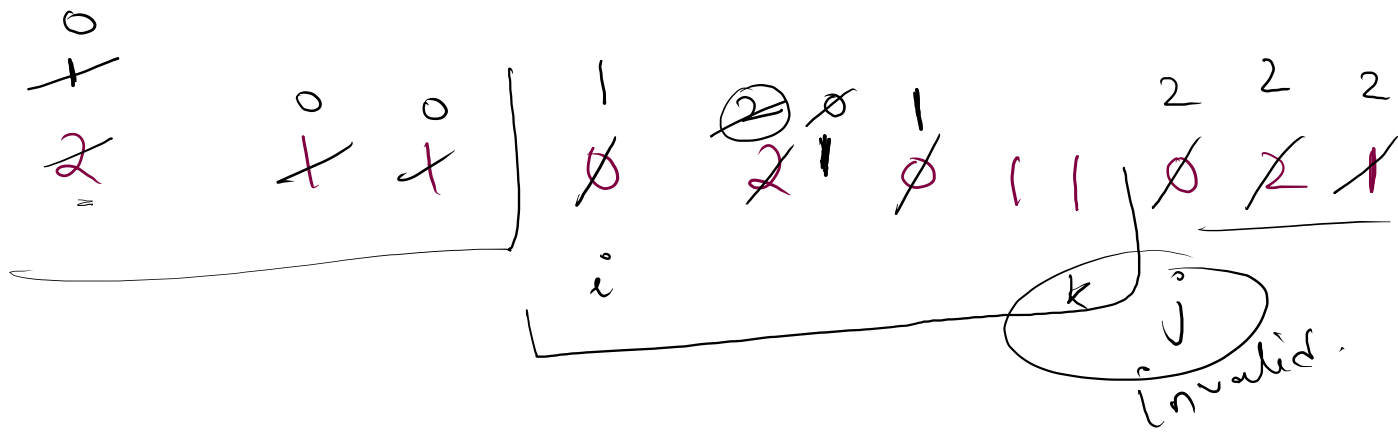
vanished
invalid.

$A[j] = 2$
 $\text{swap}(j, k)$

$k--$

$A[j] = j$
 $\text{swap}(i, j)$
 $j++$
 $i++$

$A[j] = 1$
 $j++$



$A[j] = 0 \rightarrow \text{swap}(i, j) \quad i++ \quad j++$

$A[j] = 1 \rightarrow j++$

$A[j] = 2 \rightarrow \text{swap}(j, k) \quad k--$


```

1  class Solution {
2
3  public void swap(int x, int y, int [] A){
4      int tmp = A[x];
5      A[x] = A[y];
6      A[y] = tmp;
7  }
8  public void sortColors(int[] A) {
9      int i = 0;
10     int j = 0;
11     int k = A.length-1;
12
13
14     while(j <= k){
15         if (A[j] == 0){
16             swap(i, j, A);
17             i++;
18             j++;
19
20         }else if(A[j] == 1){
21             j++;
22
23         }else{
24             swap(j, k, A);
25             k--;
26         }
27     }
28 }
29

```

$$0 \rightarrow i-1 = 0$$

$$i \rightarrow j-1 = 1$$

$$k+1 \rightarrow n-1 = 2$$

j to k → unknown

Reach Target

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 pointer**. answer must be **unique**.

$n=6$
 $target = 4$

Sample Input 0

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

-1	1	2	3	4	5
0	1	2	3	4	5

Sample Output 0

```
0 5
1 3
```

i
 j

0 5
1 3

$s == t$
↳ print
 $i++ j--$

$s > t$
 $j--$

$s < t$
 $i++$

$$\underline{\text{tar} = 4.}$$

```
1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner scn = new Scanner(System.in);
8         int n = scn.nextInt();
9         int [] A = new int[n];
10        for(int i = 0; i < n; i++){
11            A[i] = scn.nextInt();
12        }
13        int tar = scn.nextInt();
14
15        int i = 0;
16        int j = n-1;
17
18        while(i < j){
19            int sum = A[i] + A[j];
20            if(sum == tar){
21                System.out.println(i + " " + j);
22                i++;
23                j--;
24            }
25            else if(sum > tar){
26                j--;
27            }else if(tar > sum){
28                i++;
29            }
30        }
31    }
32 }
```

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

i
j

0 7
1 5
2 4