Interleaving x and y Elements

$n = 10$

$$arr = \begin{array}{c} \begin{array}{cccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array} \\ \left[\underbrace{x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5}_{} \quad \underbrace{y_1 \quad y_2 \quad y_3 \quad y_4 \quad y_5}_{} \right] \\ \begin{array}{cc} \uparrow & \uparrow \\ i & j \end{array} \end{array}$$

$$ans = \left[\underbrace{x_1 \quad y_1}_{} \quad \underbrace{x_2 \quad y_2}_{} \quad \underbrace{x_3 \quad y_3}_{} \quad \underbrace{x_4 \quad y_4}_{} \quad \underbrace{x_5 \quad y_5}_{} \right]$$

$\uparrow$

k

$$\text{int } i = 0, \quad j = \frac{n}{2};$$

$$\boxed{n = 10}$$

	0	1	2	3	4	5	6	7	8	9
arr =	1	2	3	4	5	6	7	8	9	10

i $\qquad$ j

$$\text{ans} = [\ 1\ 6\ \ 2\ 7\ \ 3\ 8\ \ 4\ 9\ \ 5\ 10\ ]$$

k

Code

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];
    for (int i = 0; i < n; i++) {
        arr[i] = scn.nextInt();
    }
    int[] ans = interleavingXY(arr, n);
    for (int i = 0; i < n; i++) {
        System.out.print(ans[i] + " ");
    }
}
public static int[] interleavingXY(int[] arr, int n) {
    int[] ans = new int[n];
    int i = 0;
    int j = n / 2;
    int k = 0;
    while ( k < ans.length ) {
        ans[k] = arr[i];
        i++;
        k++;

        ans[k] = arr[j];
        k++;
        j++;
    }
    return ans;
}
```

T.C = O(n)

where, n is size of array

S.C = O(n)

Rotate Right (Imp)

$n = 7$

arr =

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

$K = 2$

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

$n = 7$

	0	1	2	3	4	5	6
arr =	1	2	3	4	5	6	7

$K = 2$

trick

magic

Step1 reverse K elements from last

	0	1	2	3	4	5	6
arr =	1	2	3	4	5	7	6

$(n-K, n-1)$

Step2 reverse rest of the elements

	0	1	2	3	4	5	6
arr =	5	4	3	2	1	7	6

$(0, n-K-1)$

Step3 reverse entire array

	0	1	2	3	4	5	6
arr =	6	7	1	2	3	4	5

$(0, n-1)$

$$arr = \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \hline \end{array}$$

$$arr = [\,0 \quad -2 \quad 3 \quad 5 \quad 9 \quad 8 \quad 9 \quad 4\,]$$

$k = 4$

$n = 8$

$0 \qquad (n-k-1) \qquad (n-k) \qquad (n-1)$

$0 \qquad (n-1)$

Explanation for $k = k \% n$

arr [1 2 3 4 5]

k=1, 5 1 2 3 4
k=2, 4 5 1 2 3
k=3, 3 4 5 1 2
k=4, 2 3 4 5 1
k=5, 1 2 3 4 5
k=6, 5 1 2 3 4
k=7, 4 5 1 2 3

n=5

k=1, 6, 11, 16, 21, 26, ...
k=2, 7, 12, 17, 22, 27, ...
k=3, 8, 13, ...
k=4, 9, 14, ...
k=5, 10, 15, ...

$\boxed{k = k \% n;}$

k = 50 % 5;

k = 3 % 5 = 3

k = 26

k = 1

26 % 5 = 1

$\boxed{k = 0}$

Code

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];
    for (int i = 0; i < n; i++) {
        arr[i] = scn.nextInt();
    }
    int k = scn.nextInt();
    rotateByK(arr, n, k);
}
public static void rotateByK(int[] arr, int n, int k) {
    k = k % n;

    reverse(arr, n - k, n - 1);
    reverse(arr, 0, n - k - 1);
    reverse(arr, 0, n - 1);

    // print
    for (int i = 0; i < n; i++) {
        System.out.print(arr[i] + " ");
    }
}
public static void reverse(int[] arr, int i, int j) {
    while (i < j) {
        swap(arr, i, j);
        i++;
        j--;
    }
}
public static void swap(int[] arr, int x, int y) {
    int temp = arr[x];
    arr[x] = arr[y];
    arr[y] = temp;
}
```

T.C = O(n)

n is size of array

S.C = O(1)

Zeroes and Ones

arr =

0	1	2	3	4	5	6	7	8	9	10	11
0	1	0	0	1	1	0	1	0	1	1	0

arr =

0	1	2	3	4	5	6	7	8	9	10	11
0	0	0	0	0	0	1	1	1	1	1	1

1) Arrays.sort(arr) $O(n\log(n))$

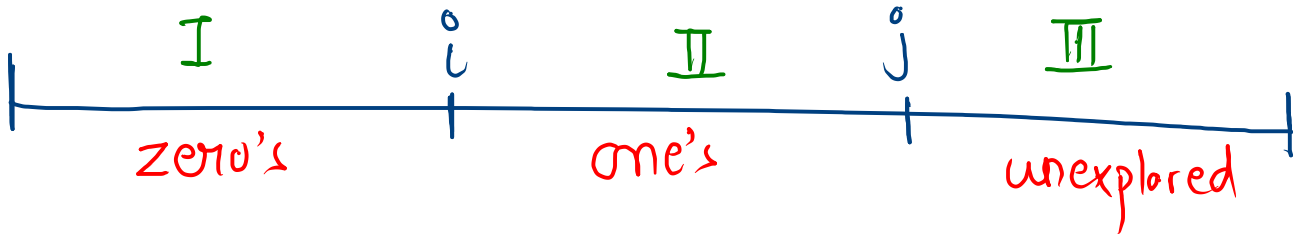
2) Only 1 time traversal is allowed

arr =

0	1	2	3	4	5	6	7	8	9	10	11
0	0	0	0	0	0	1	1	1	1	1	1

i j

Faith:-



psudo

```
if (arr[j] == 1){
        j++;
} else {
        swap(i,j)
        i++; j++;
}
```

Code

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];
    for (int i = 0; i < n; i++) {
        arr[i] = scn.nextInt();
    }
    sort01(arr, n);
}
public static void sort01(int[] arr, int n) {
    int i = 0;
    int j = 0;
    while ( j < n ) {
        if ( arr[j] == 1 ) {
            j++;
        } else {
            swap(arr, i, j);
            i++;
            j++;
        }
    }

    // print
    for (int k = 0; k < n; k++) {
        System.out.print(arr[k] + " ");
    }
}
public static void swap(int[] arr, int x, int y) {
    int temp = arr[x];
    arr[x] = arr[y];
    arr[y] = temp;
}
```

T.C = O(n)

S.C = O(1)