

Lesson:



Problems on Array-2



Pre-Requisites

- Arrays in Java

List of Concepts Involved

- Basic problem-solving in Arrays

Problem 1: Given 2 integers a and b. Swap the 2 given values.

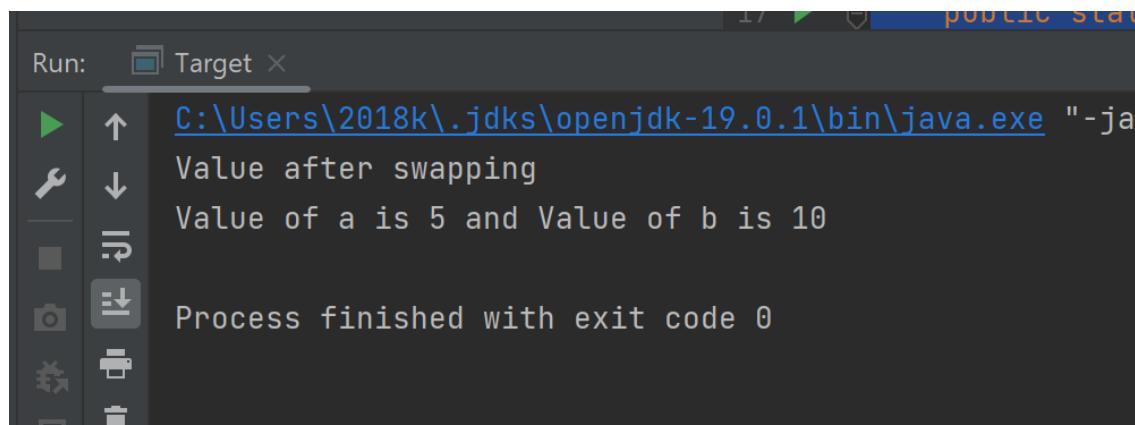
Input : a = 5, b = 10

Output : a = 10, b = 5

Method 1: Using temporary variable

Code:

```
import java.io.*;
import java.util.*;
public class Target {
    public static void swap(int a, int b)
    {
        int temp = a;
        a = b;
        b = temp;
        System.out.println("Value after swapping");
        System.out.println("Value of a is " + a
            + " and Value of b is " + b);
    }
    public static void main(String[] args){
        swap(10,5);
    }
}
```



Explanation: Create a temporary variable 'temp' and assign 'temp' as 'a'. This way the value of 'a' will be stored. Now assign 'a' the value of 'b' and b to 'temp' value. The values will be swapped.

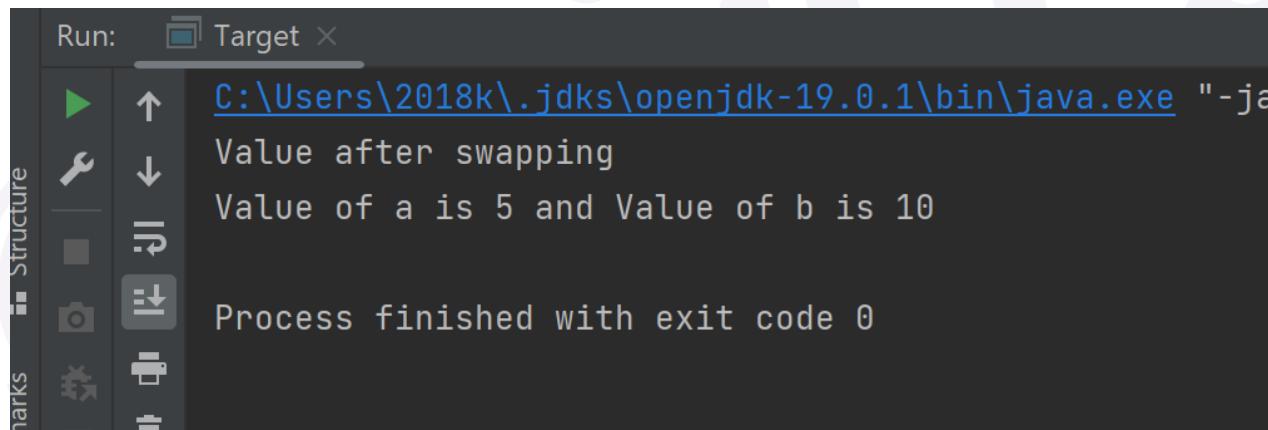
Method 2: Using Sum and Difference

Code:

```

import java.io.*;
import java.util.*;
public class Target {
    public static void swap(int a, int b)
    {
        a = a + b;
        b = a - b;
        a = a - b;
        System.out.println("Value after swapping");
        System.out.println("Value of a is " + a
            + " and Value of b is " + b);
    }
    public static void main(String[] args){
        swap(10,5);
    }
}

```



```

Run: Target ×
C:\Users\2018k\.jdks\openjdk-19.0.1\bin\java.exe "-javaagent:D:\Program Files\Java\VisualVM\lib\visualvm-agent.jar" -Dfile.encoding=UTF-8 Target
Value after swapping
Value of a is 5 and Value of b is 10
Process finished with exit code 0

```

Explanation: First we make a to be the sum of a and b. Then now we have the sum, we try to make b equal to a. We can do it by taking the total sum and subtracting b from it. This will be $a-b$. This way now b has value of a. We now make a as $a-b$, that is the total sum - value of a. This way a will have value of b, and b will have value of a.

Problem 2: Reverse the given array 'a' consisting of integer values.

Input : [1,2,3,4,5]

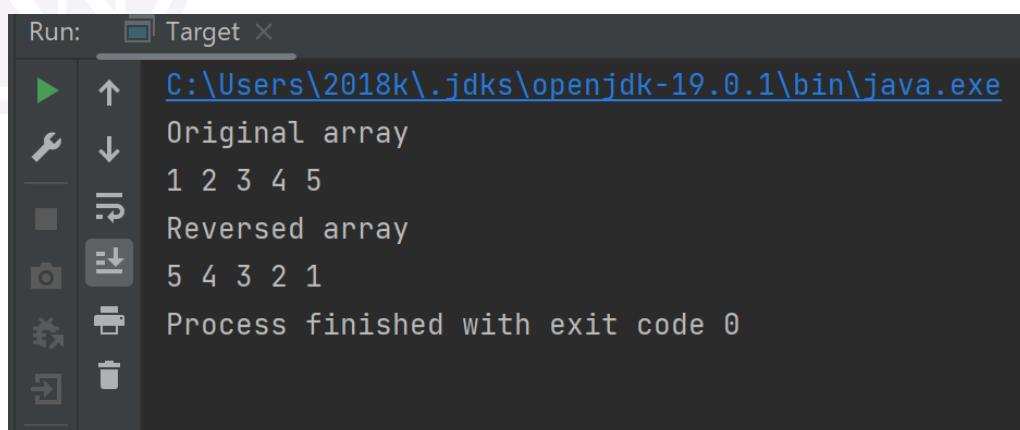
Output : [5,4,3,2,1]

Code:

```

import java.io.*;
import java.util.*;
public class Target {
    public static void reverse(int[] a, int n){
        int tmp = 0;
        int i = 0, j = n-1;
        while(i < j){
            tmp = a[i];
            a[i] = a[j];
            a[j] = tmp;
            i++;
            j--;
        }
    }
    public static void main(String[] args){
        int a[]={1,2,3,4,5};
        System.out.println("Original array ");
        for(int i=0;i<a.length;i++){
            System.out.print(a[i]+" ");
        }
        System.out.println("");
        int n=a.length;
        reverse(a,n);
        System.out.println("Reversed array ");
        for(int i=0;i<a.length;i++){
            System.out.print(a[i]+" ");
        }
    }
}

```



Explanation: Take two pointers one from start and another from end and keep on swapping the values. After swapping the values, increment the starting pointer and decrement the ending pointer. Keep on repeating this process till the end pointer is greater than the starting pointer. The loop will break when both pointers become equal or the starting pointer exceeds the ending pointer.

Problem 3: Rotate the given array 'a' by k steps, where k is non-negative.

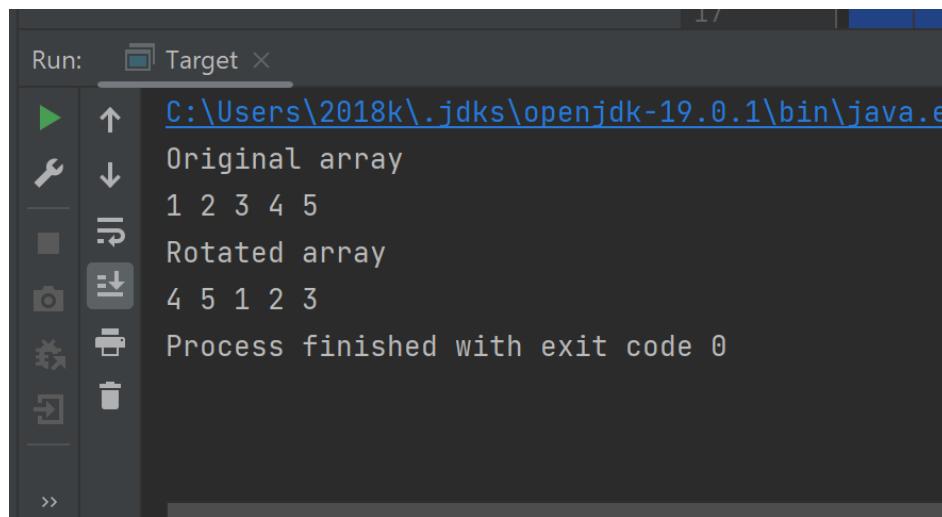
Note: k can be greater than n as well.

Input: a = [1,2,3,4,5,6,7], k = 3

Output: [5,6,7,1,2,3,4]

Code: Method 1:

```
// with extra space
import java.io.*;
import java.util.*;
public class Target {
    public static void rotate(int[] a, int k) {
        int n = a.length;
        k %= n; // k can be greater than n
        int[] ans = new int[n];
        for(int i = n-k; i <= n-1; i++) {
            ans[i-(n-k)] = a[i];
        }
        for(int i = 0; i < n-k; i++) {
            ans[i+k] = a[i];
        }
        for(int i = 0; i < n; i++) {
            a[i] = ans[i];
        }
    }
    public static void main(String[] args){
        int a[]={1,2,3,4,5};
        System.out.println("Original array ");
        for(int i=0;i<a.length;i++){
            System.out.print(a[i]+" ");
        }
        System.out.println("");
        int k=2;
        rotate(a,k);
        System.out.println("Rotated array ");
        for(int i=0;i<a.length;i++){
            System.out.print(a[i]+" ");
        }
    }
}
```



```
Run: Target ×
C:\Users\2018k\.jdks\openjdk-19.0.1\bin\java.e
Original array
1 2 3 4 5
Rotated array
4 5 1 2 3
Process finished with exit code 0
```

Explanation Method 1: Create a new answer array and add the last k elements first in the answer array and the rest of the elements after that. We can do that by traversing the array twice as shown. Note: Make sure to take modulus of k by n , as value of k can be greater than n so it will become less than n as :

$$k = \text{constant} * n + \text{reminder}$$

$$\text{Here reminder} = k \% n$$

If we rotate an array n times it will be worthless as it will come back to the same position, so just remove that from k and take the remainder part only.

Code: Method 2:

```
// without extra space
import java.util.*;
import java.lang.*;
public class Main {
    // without extra space
    public static void rotate(int[] a, int k) {
        k %= a.length;
        reverse(a, 0, a.length - 1);
        reverse(a, 0, k - 1);
        reverse(a, k, a.length - 1);
    }
    private static void reverse(int[] a, int i, int j){
        int tmp = 0;
        while(i < j){
            tmp = a[i];
            a[i] = a[j];
            a[j] = tmp;
            i++;
            j--;
        }
    }
}
```

```

public static void main(String[] args) {
    Scanner sc=new Scanner(System.in);
    int n;
    System.out.println("Enter n");
    n=sc.nextInt();
    System.out.println("Enter k");
    int k;
    k = sc.nextInt();
    System.out.println("Enter the array elements");
    int[] a = new int[n];
    for (int i = 0; i < n; i++) {
        a[i]=sc.nextInt();
    }
    System.out.println("Output: ");
    rotate(a, k);
    for(int i = 0; i < n; i++) {
        System.out.print(a[i] + " ");
    }
}
}

```

Run: Main ×

- ▶ ↑ "C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2023.2.1\lib\idea_rt.jar" -Dfile.encoding=UTF-8 Main
- 🔧 ↓ Enter n
- ➡ 7
- ➡ Enter k
- ➡ 3
- 🖨 Enter the array elements
- ➡ 1 2 3 4 5 6 7
- ➡ Output:
- ➡ 5 6 7 1 2 3 4
- ➡ Process finished with exit code 0

Explanation Method 2: Here in this approach we first reverse the whole array. Then we reverse the first k elements of the array. Then we reverse the last $n-k$ elements of the array. This way without extra space, we have rotated the array by ' k ' steps.

Let us dry run to understand this intuitive approach.

Say $n = 7$, $k = 3$ and the array = [1,2,3,4,5,6,7].

Reverse the whole array, so array = [7,6,5,4,3,2,1]

Reverse the first k elements, so array = [5,6,7,4,3,2,1]

Reverse [$k+1$, n] elements, so array = [5,6,7,1,2,3,4]

Problem 4: Given Q queries, check if the given number is present in the array or not.

Note: Value of all the elements in the array is less than 105.

Input :

```
4
1 1 2 7
4
1
2
3
4
```

Output:

```
YES
YES
NO
NO
```

Code: Method 1:

```
import java.util.*;
import java.lang.*;
public class Main {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        int n;
        n=sc.nextInt();
        int[] a = new int[n];
        for (int i = 0; i < n; i++) {
            a[i]=sc.nextInt();
        }
        int N = 100100;
        // creating a frequency array as max(a[i]) < 105
        int[] freq = new int[N];
        for (int i = 0; i < n; i++) {
            freq[a[i]] = freq[a[i]] + 1;
        }
        int q;
        q = sc.nextInt();
        while (q-- > 0) {
            int val;
            val = sc.nextInt();
            if (freq[val] > 0) {
                System.out.println("YES");
            } else System.out.println("NO");
        }
    }
}
```

```
Run: Main ×
C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Ed
4
1 1 2 7
4
1
2
3
4
YES
YES
NO
NO

Process finished with exit code 0
```

Explanation : Create a frequency array and just increment the count of values the user is entering. This way we have the count of all the values the user has entered and the answer would be YES wherever the count of the values will be positive and wherever the count will be 0, the answer will be NO.

Upcoming Class Teasers:

- Problem Solving in Arrays