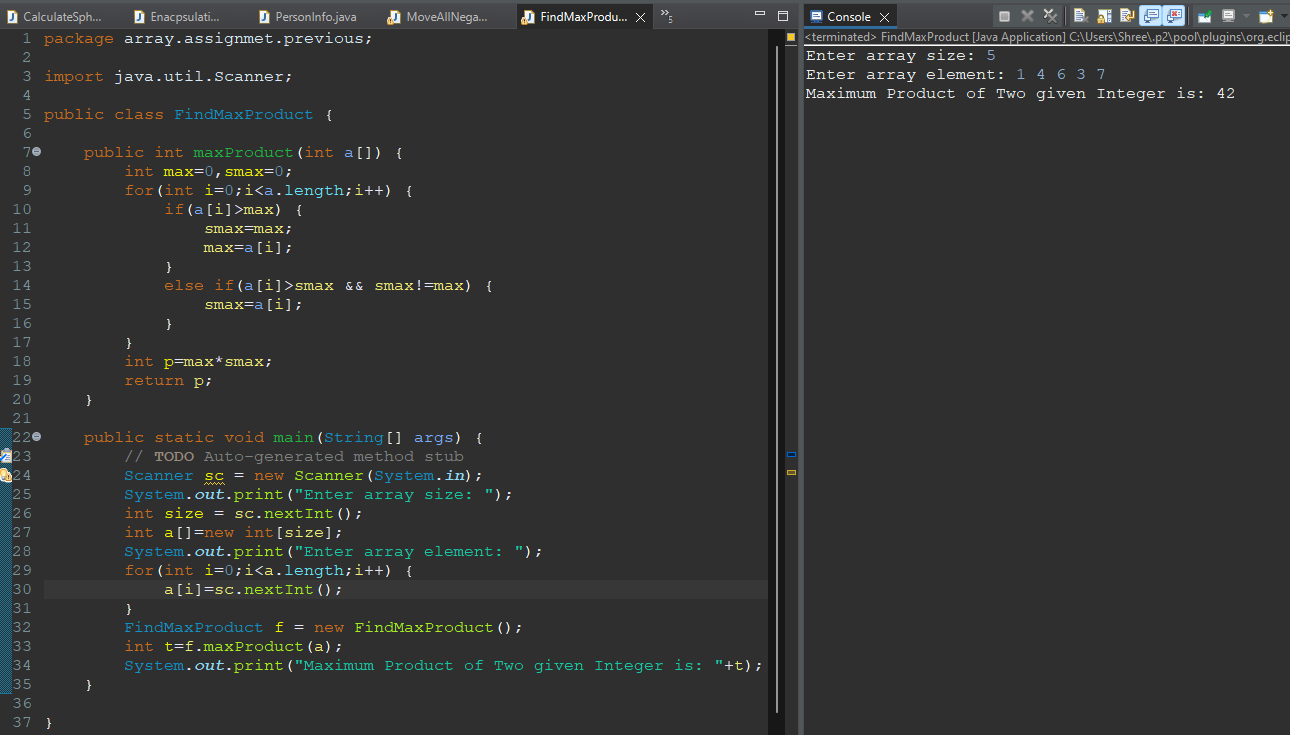
**Assignment No:-28**

Name:-Suryawanshi Sangramsingh Sambhaji

Batch: - Delta - DCA (Java) 2024 Date:-13/6/2024

**1. Write a Java program to find maximum product of two integers in a given array of integers.**

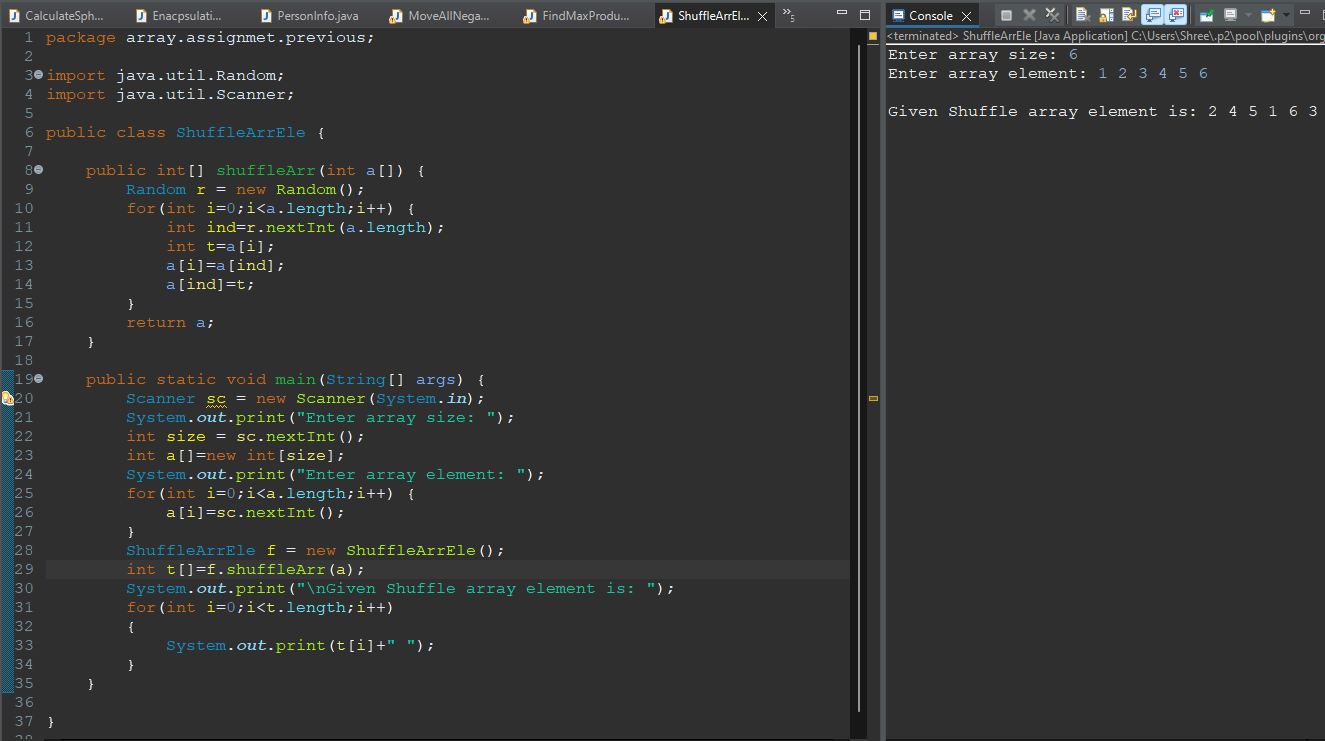
****

**2. Write a Java program to shuffle a given array of integers.**

**Example:**

**Input : nums = { 1, 2, 3, 4, 5, 6 }**

**Output: Shuffle Array: [4, 2, 6, 5, 1, 3]**

****

**3. Write a Java program to rearrange a given array of unique elements such that every second element of the array is greater than its left and right elements.**

**Example:**

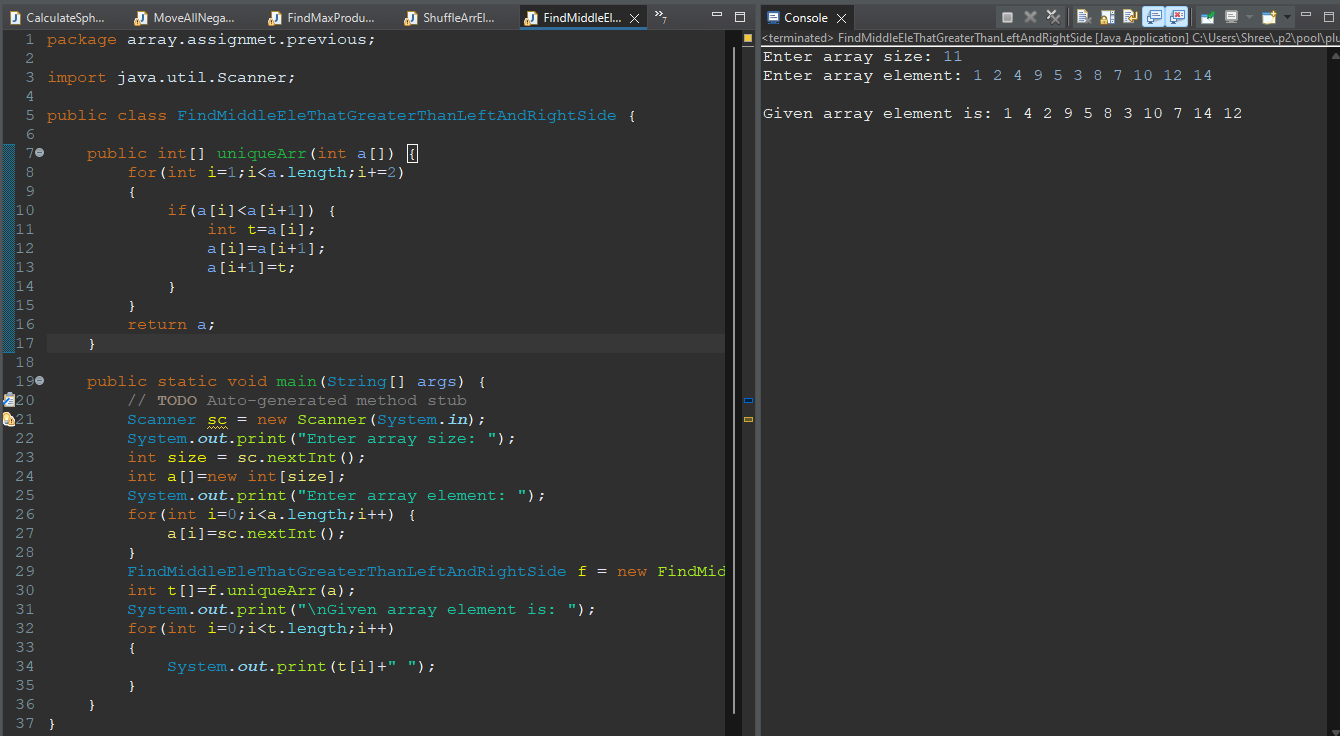
**Input :**

**nums= { 1, 2, 4, 9, 5, 3, 8, 7, 10, 12, 14 }**

**Output:**

**Array with every second element is greater than its left and right elements:**

**[1, 4, 2, 9, 3, 8, 5, 10, 7, 14, 12]**

****

**4. Write a Java program to replace each element of the array with product of every other element in a given array of integers.**

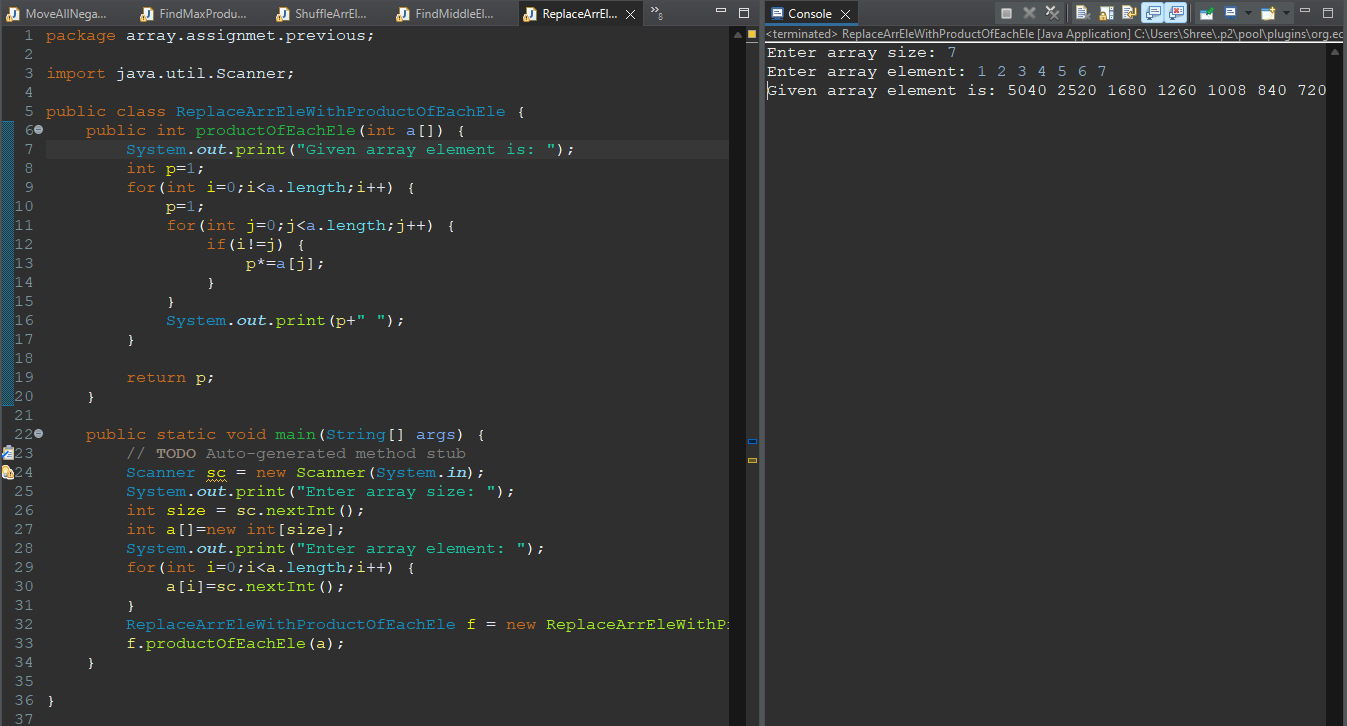
**Example:**

**Input : nums1 = { 1, 2, 3, 4, 5, 6, 7}**

**Output:**

**Array with product of every other element:**

**[5040, 2520, 1680, 1260, 1008, 840, 720]**

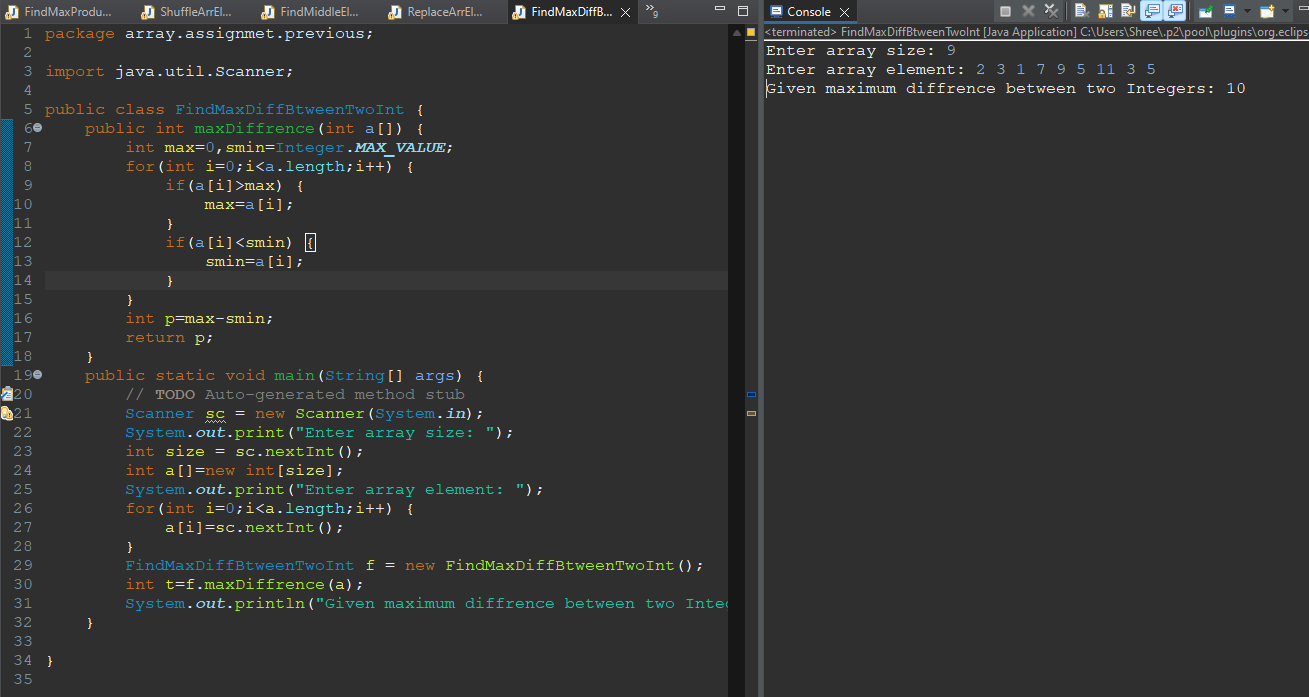
****

**5. Write a Java program to find maximum difference between two elements in a given array of integers such that smaller element appears before larger element.**

**Example:**

**Input : nums = { 2, 3, 1, 7, 9, 5, 11, 3, 5 }**

**Output:The maximum difference between two elements of the said array elements:10**

****

**6. Find a peak element which is not smaller than its neighbours**

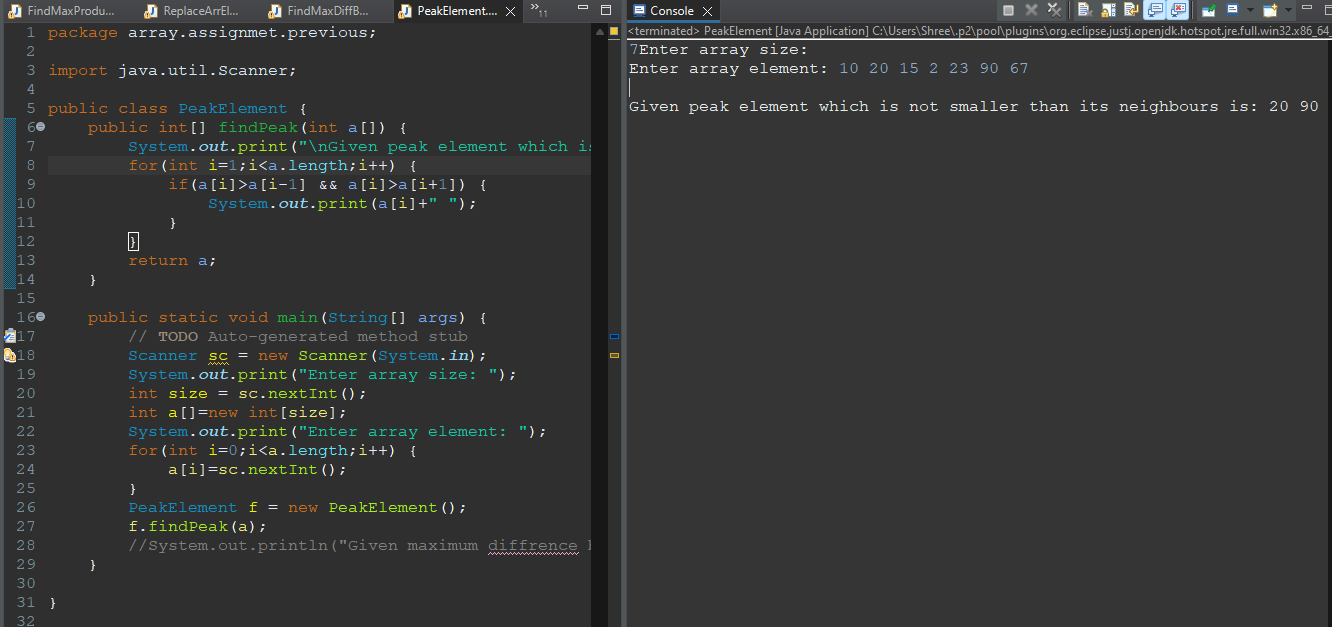
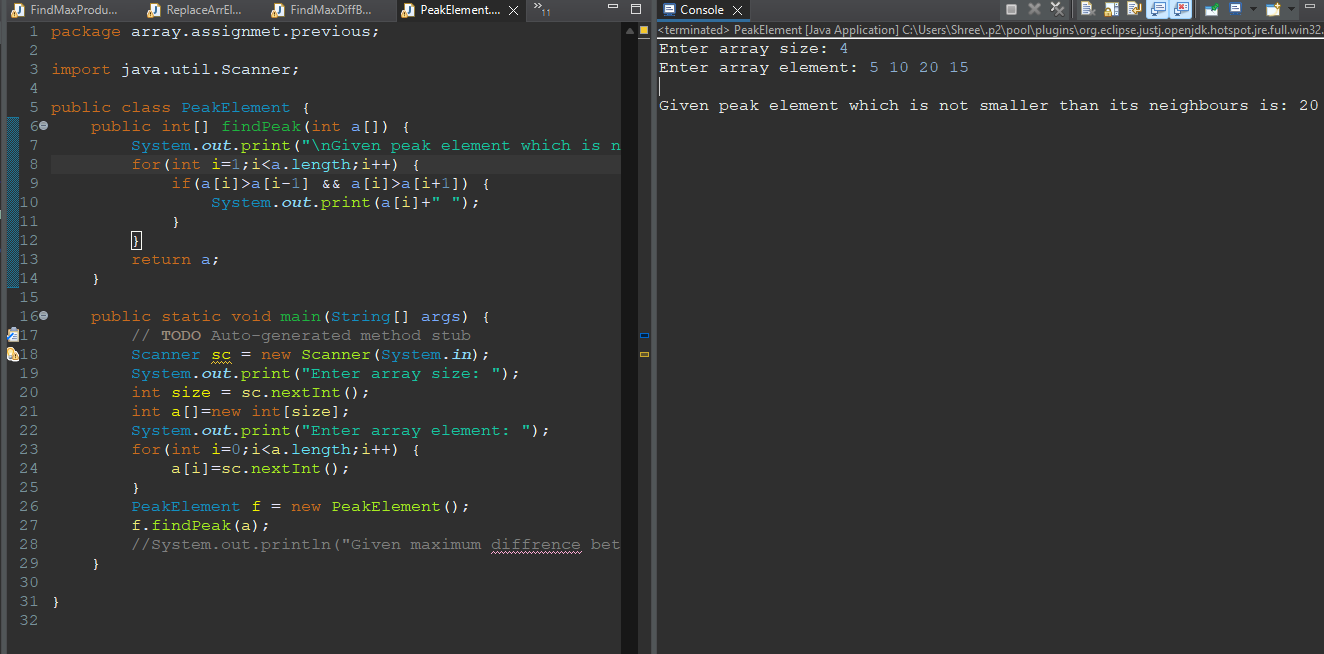
**Input: array[]= {5, 10, 20, 15} Output: 20**

**Explanation: The element 20 has neighbors 10 and 15, both of them are less than 20.**

**Input: array[] = {10, 20, 15, 2, 23, 90, 67}**

**Output: 20 or 90**

**Explanation: The element 20 has neighbors 10 and 15, both of them are less than 20, similarly 90 has neighbors 23 and 67.**

****

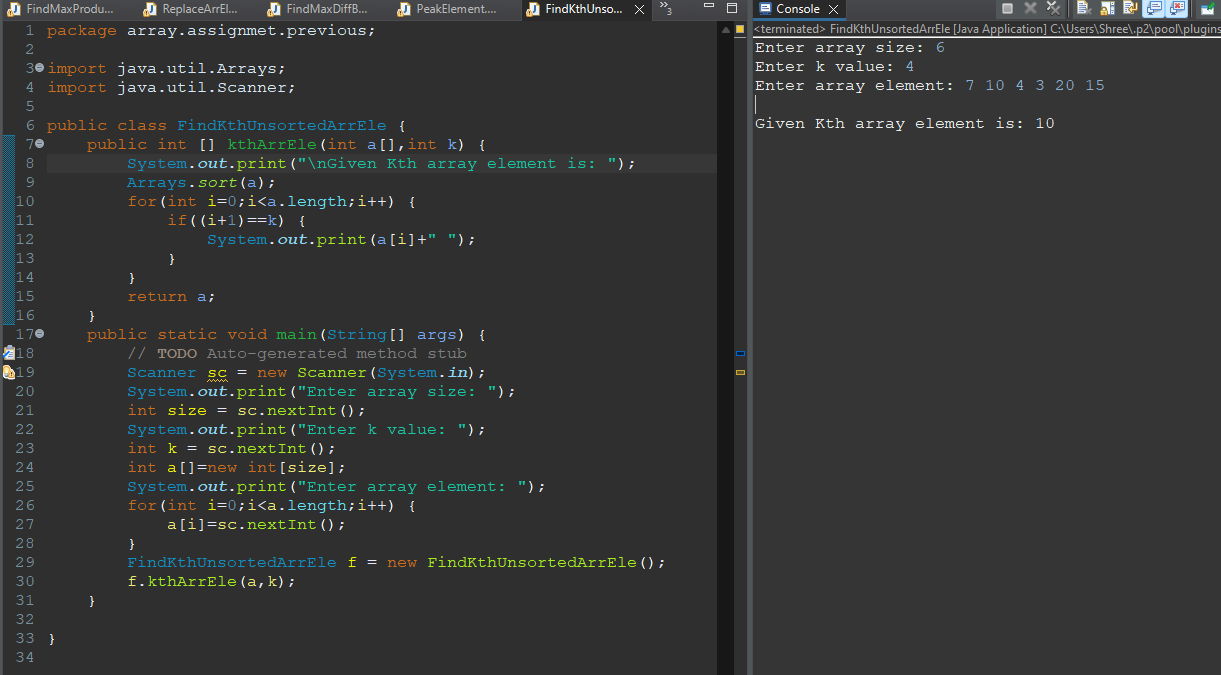
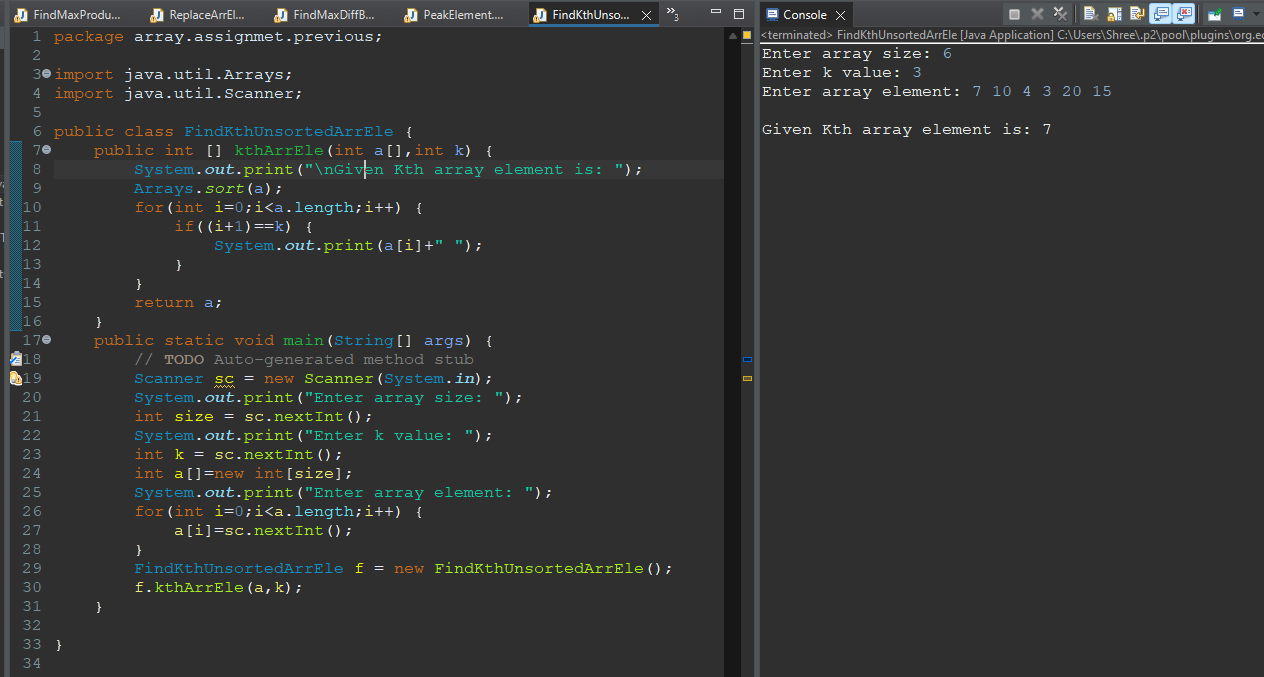
**7.K’th Largest Element in Unsorted Array**

**Input: arr[] = {7, 10, 4, 3, 20, 15}, K = 3**

**Output: 7**

**Input: arr[] = {7, 10, 4, 3, 20, 15}, K = 4**

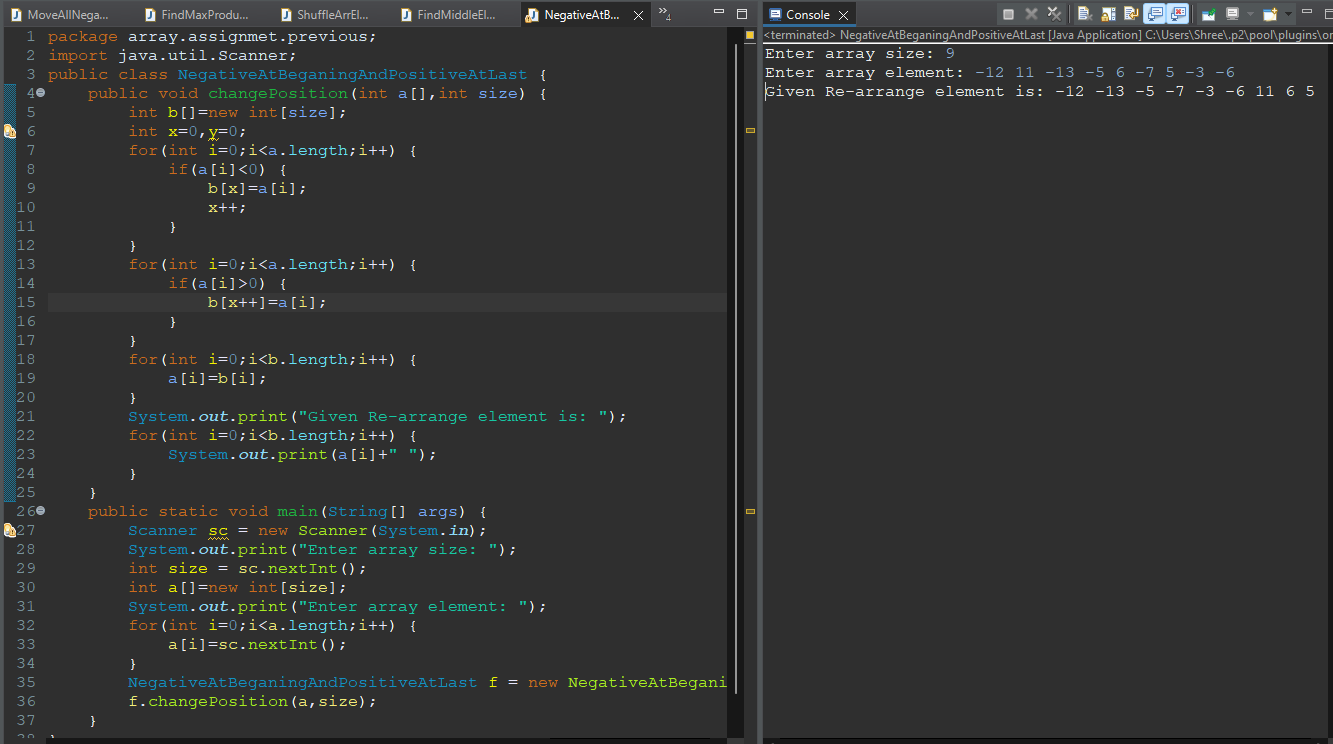
**Output: 10**

****

**8. Move all negative numbers to beginning and positive to end with constant extra space**

**Input: -12, 11, -13, -5, 6, -7, 5, -3, -6**

**Output: -12 -13 -5 -7 -3 -6 11 6 5**

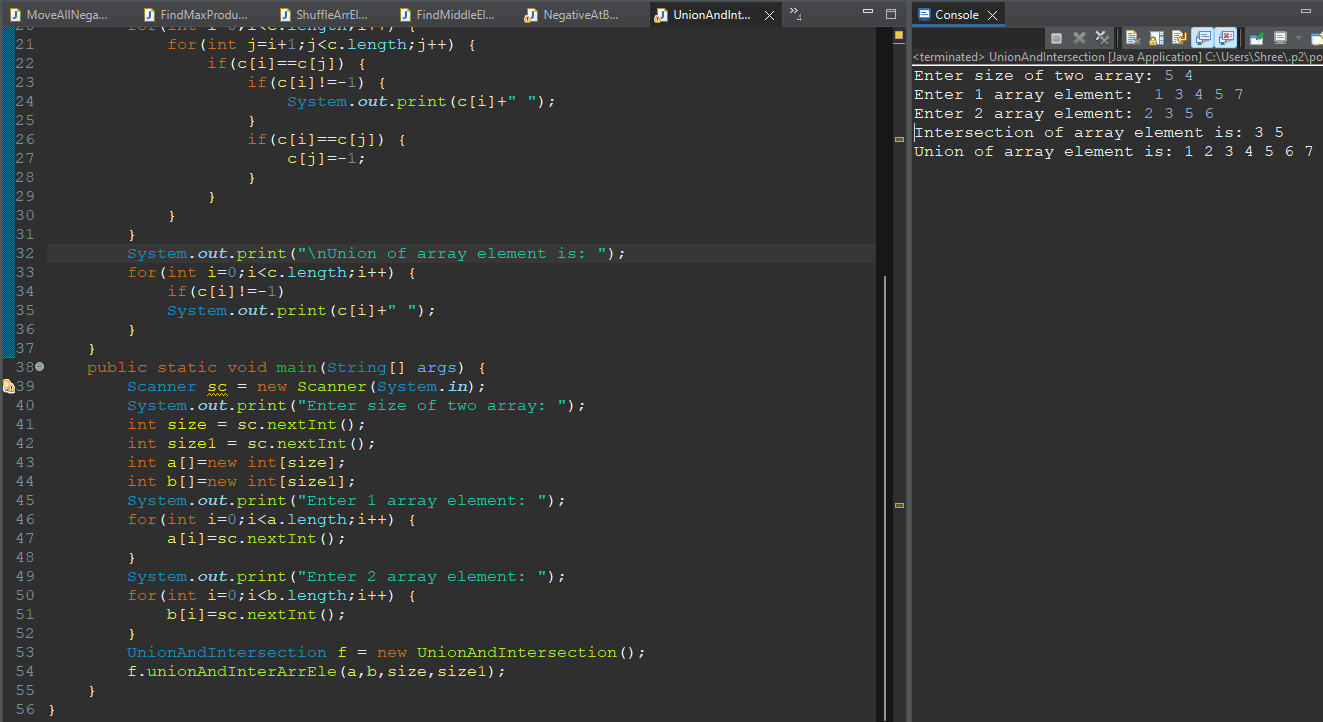
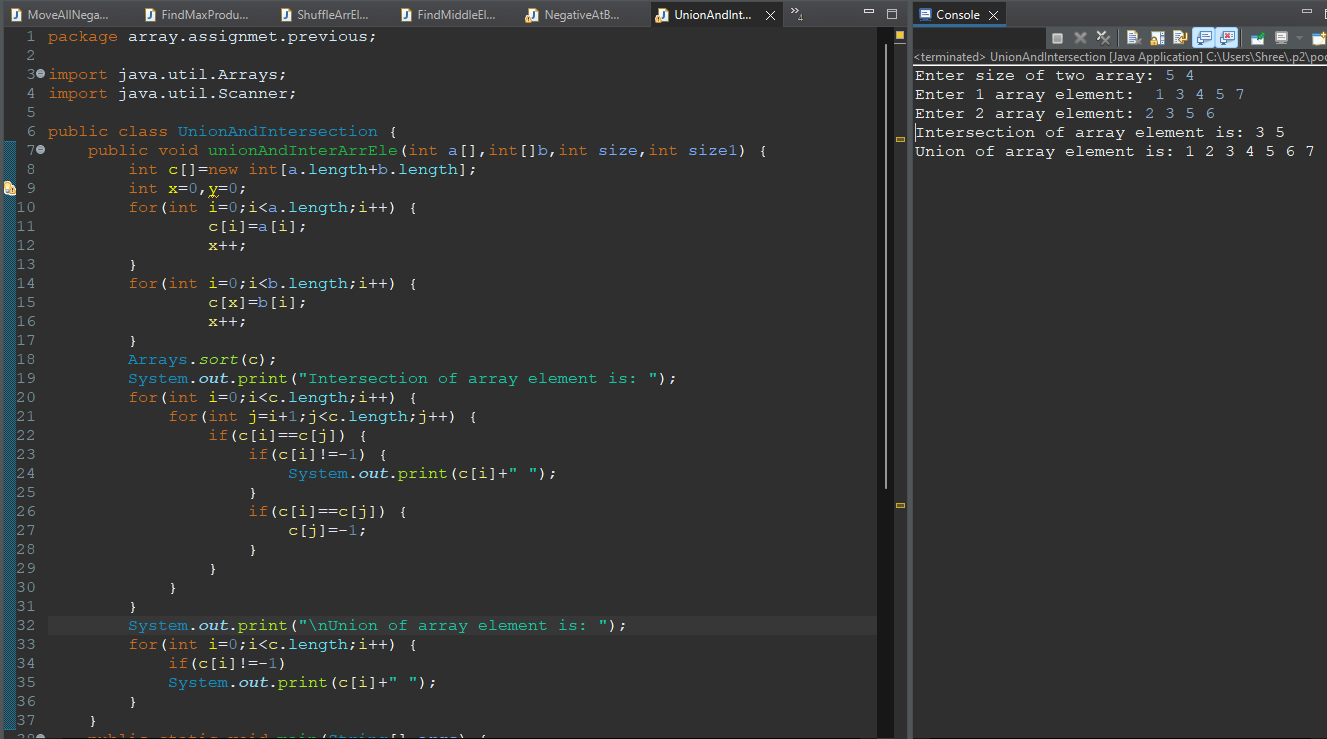
****

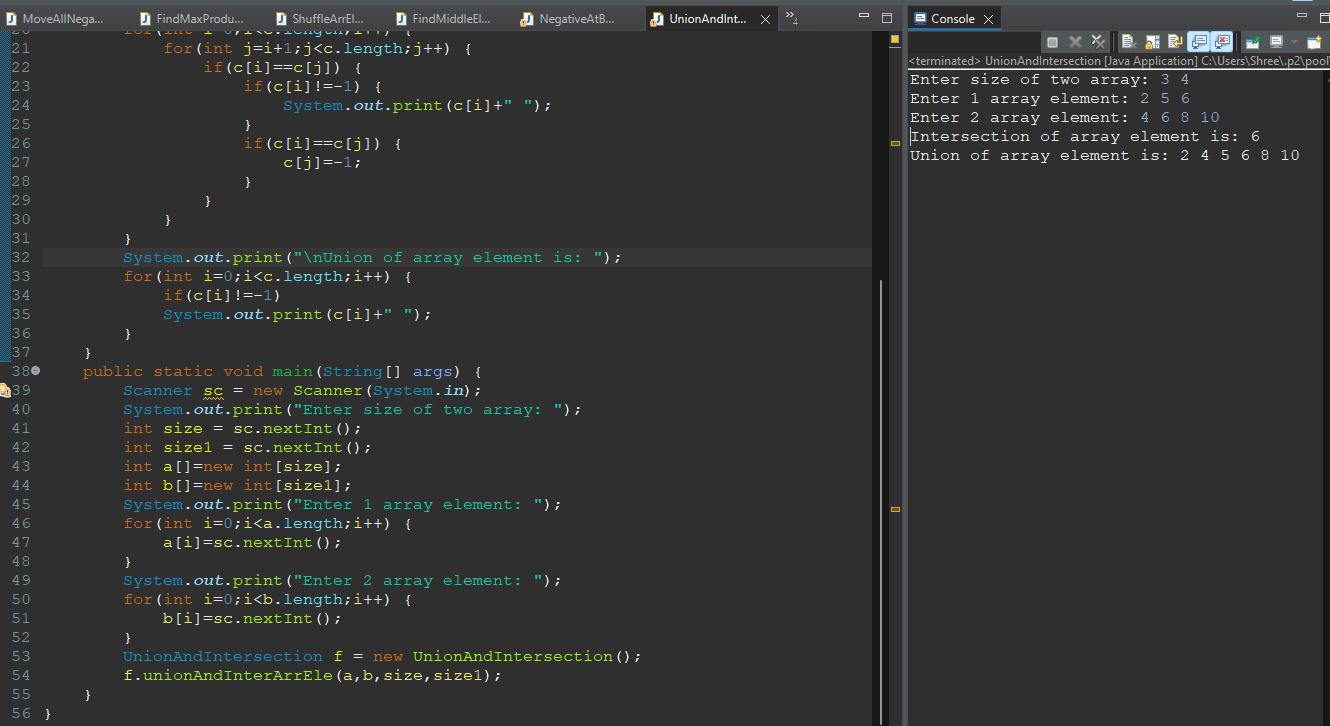
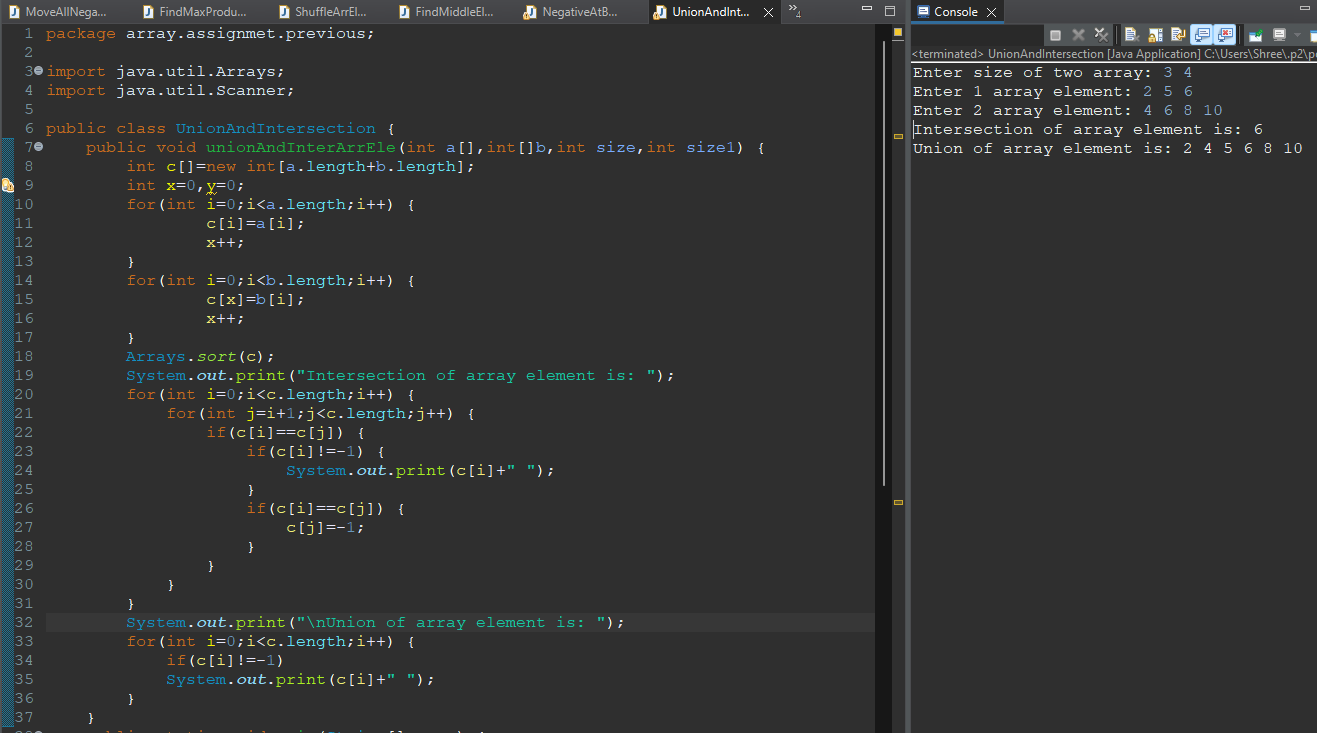
**Q9.Union and Intersection of two sorted arrays**

**Input: arr1[] = {1, 3, 4, 5, 7} arr2[] = {2, 3, 5, 6} Output: Union: {1, 2, 3, 4, 5, 6, 7} Intersection: {3, 5}**

**Input: arr1[] = {2, 5, 6}arr2[] = {4, 6, 8, 10}**

**Output: Union : {2, 4, 5, 6, 8, 10} Intersection : {6}**

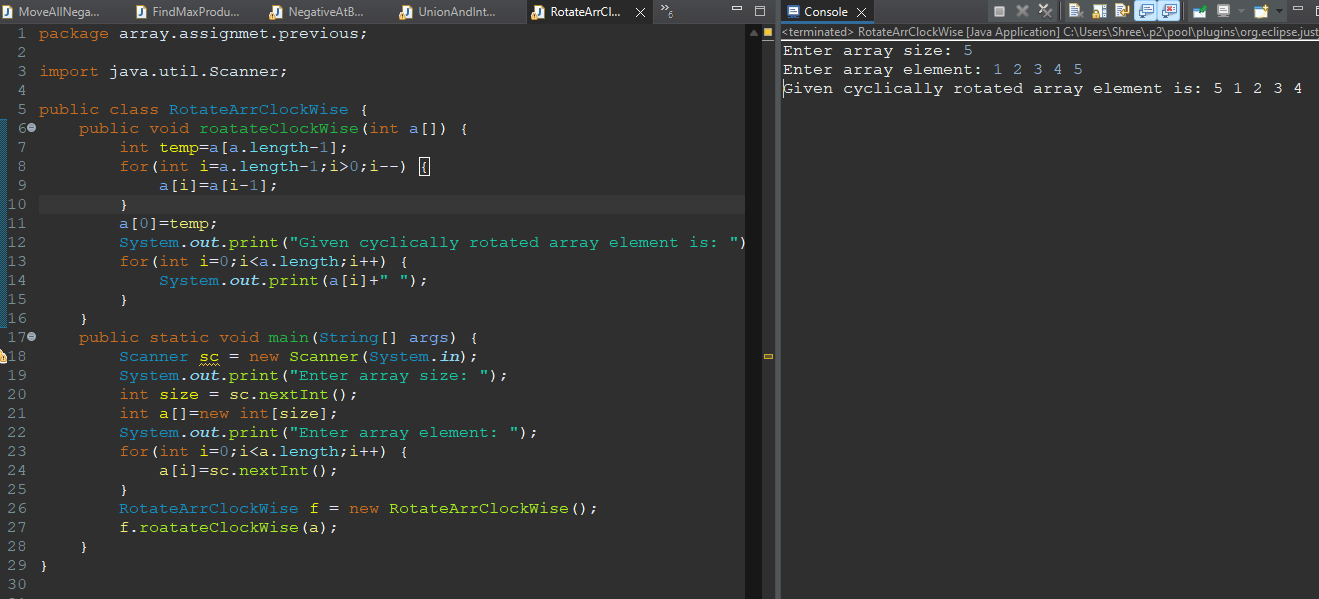
****

****

**Q10. Program to cyclically rotate an array by one**

**Input: arr[] = {1, 2, 3, 4, 5}**

**Output: arr[] = {5, 1, 2, 3, 4}**

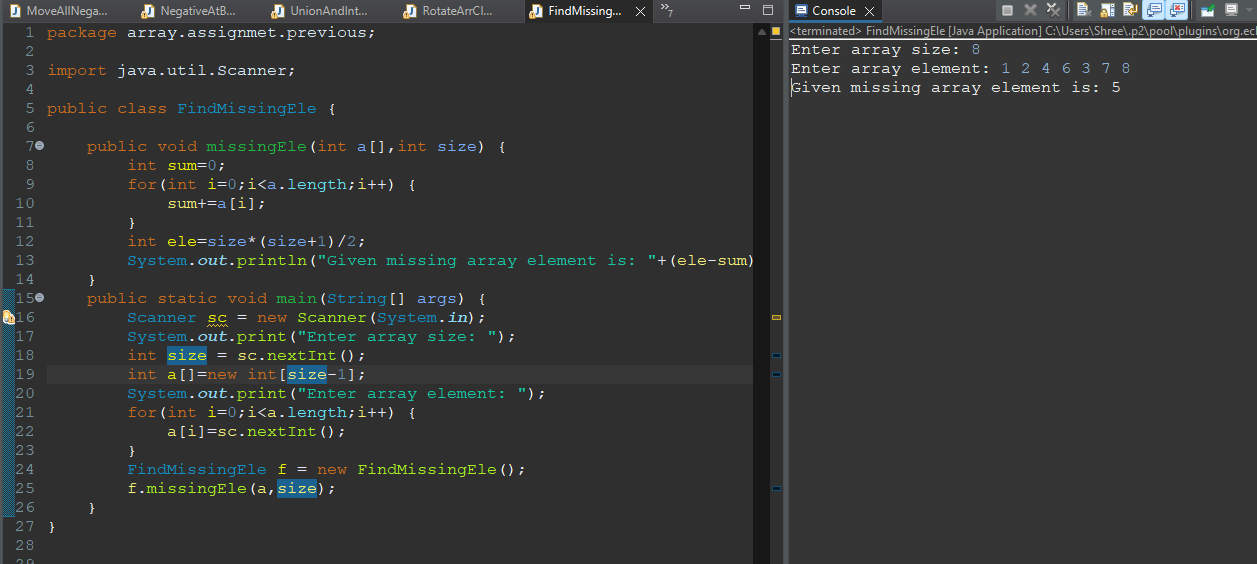
****

**11. Find the Missing Number**

**Input: arr[] = {1, 2, 4, 6, 3, 7, 8}, N = 8**

**Output: 5**

**Explanation: The missing number between 1 to 8 is 5**

****

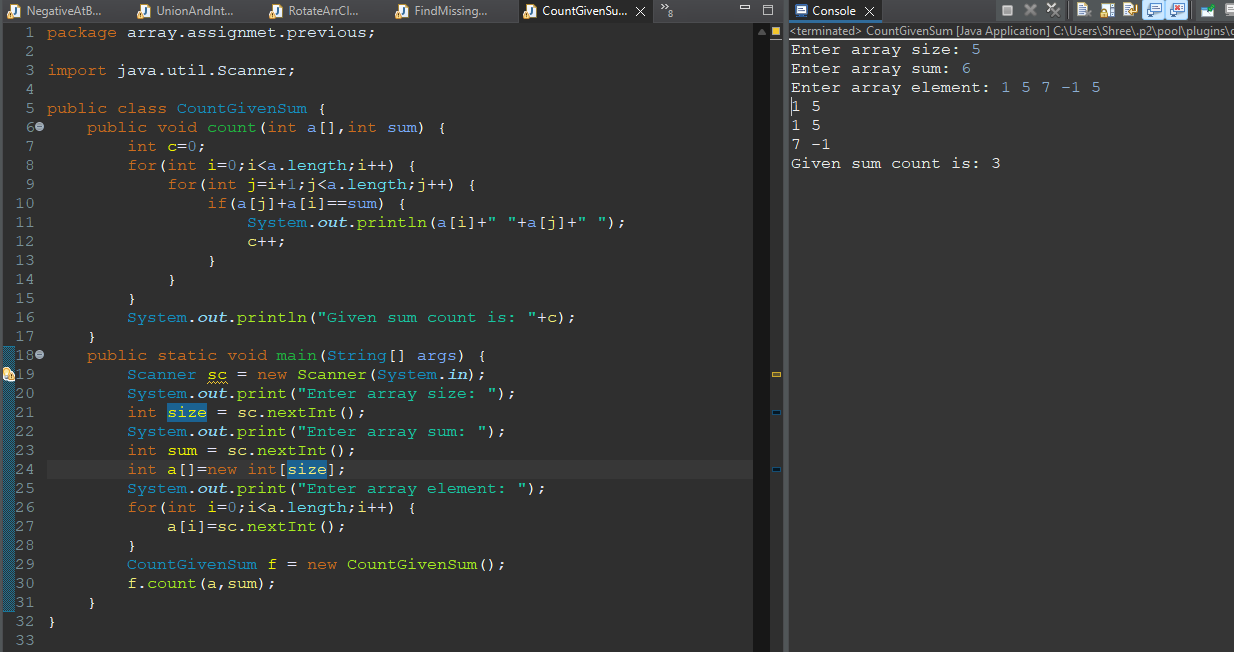
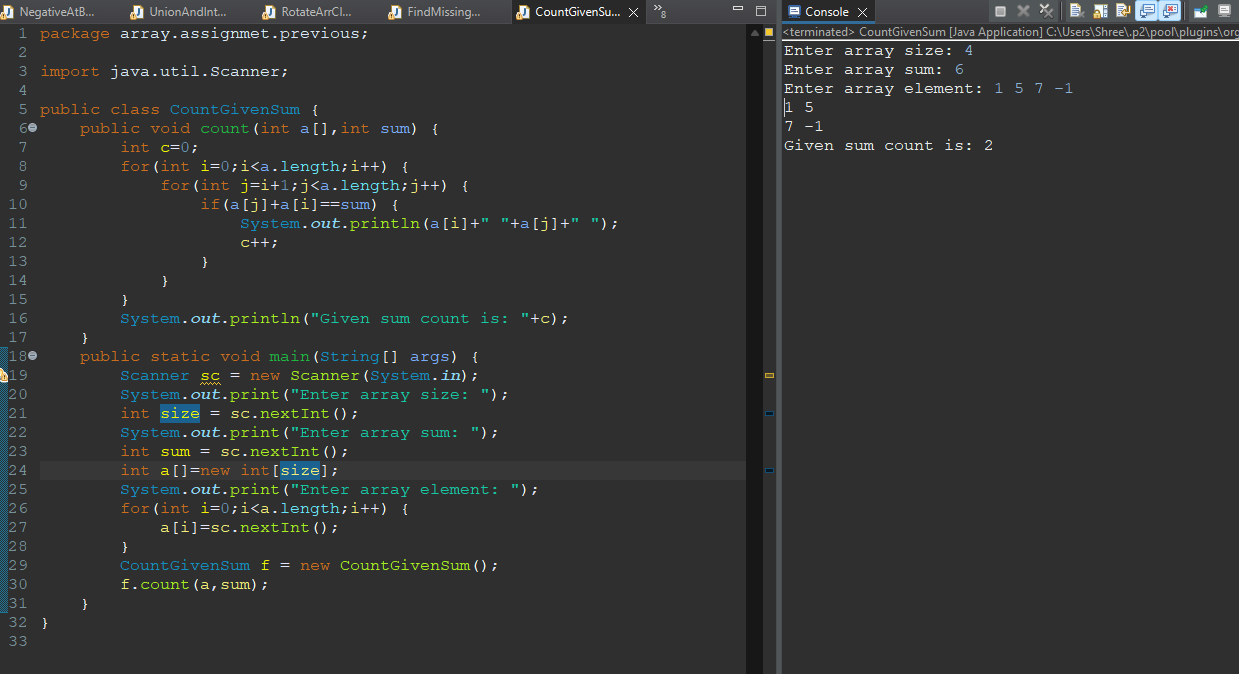
**12. Count pairs with given sum**

**nput: arr[] = {1, 5, 7, -1}, sum = 6**

**Output: 2Explanation: Pairs with sum 6 are (1, 5) and (7, -1).**

**Input: arr[] = {1, 5, 7, -1, 5}, sum = 6**

**Output: 3 Explanation: Pairs with sum 6 are (1, 5), (7, -1) & (1, 5).**

****

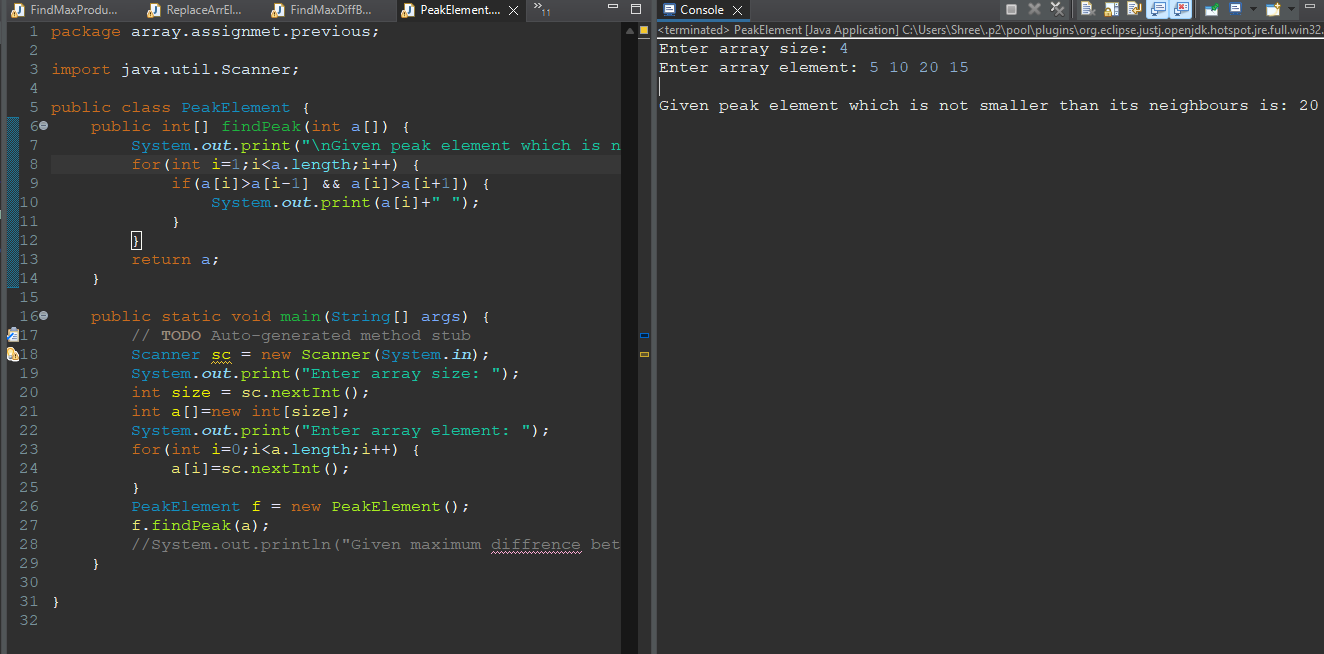
**Q1.Given an array arr[] of integers. Find a peak element i.e. an element that is not smaller than its neighbors.**

**Note: For corner elements, we need to consider only one neighbor.**

**Example:**

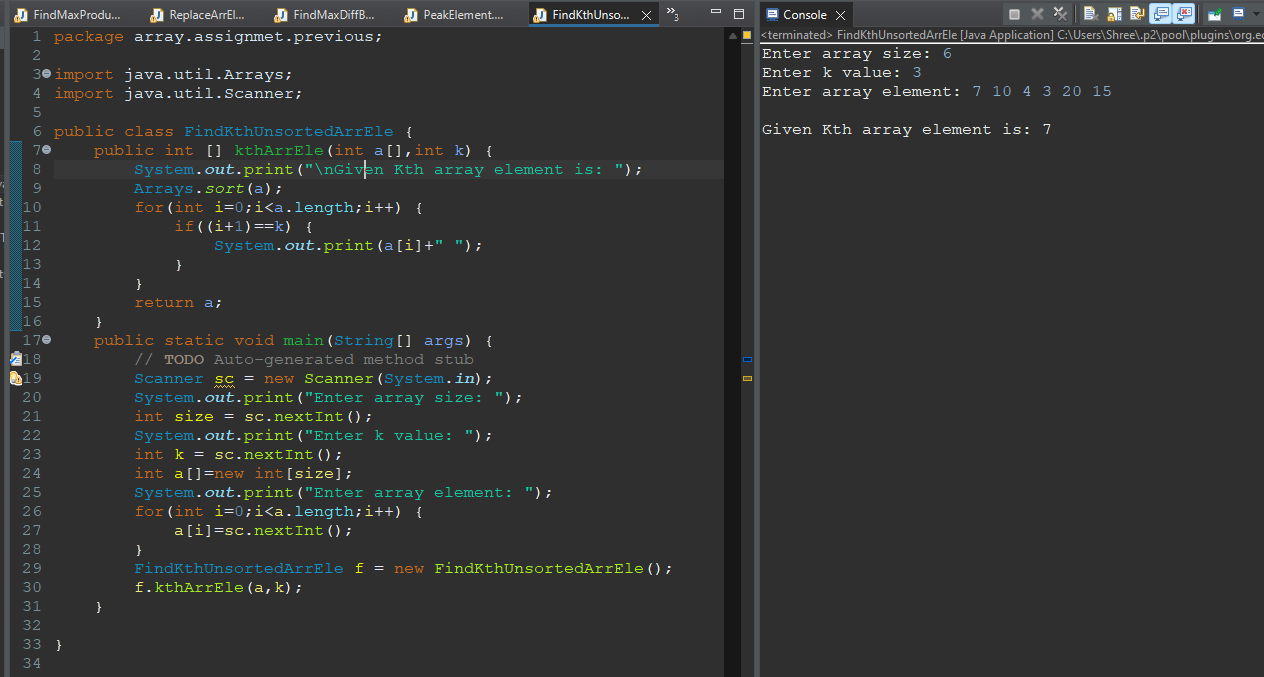
**Input: array[]= {5, 10, 20, 15}**

**Output: 20**

****

**Q2.Given an array and a number K where K is smaller than the size of the array. Find the K’th smallest element in the given array. Given that all array elements are distinct. Examples: Input: arr[] = {7, 10, 4, 3, 20, 15}, K = 3**

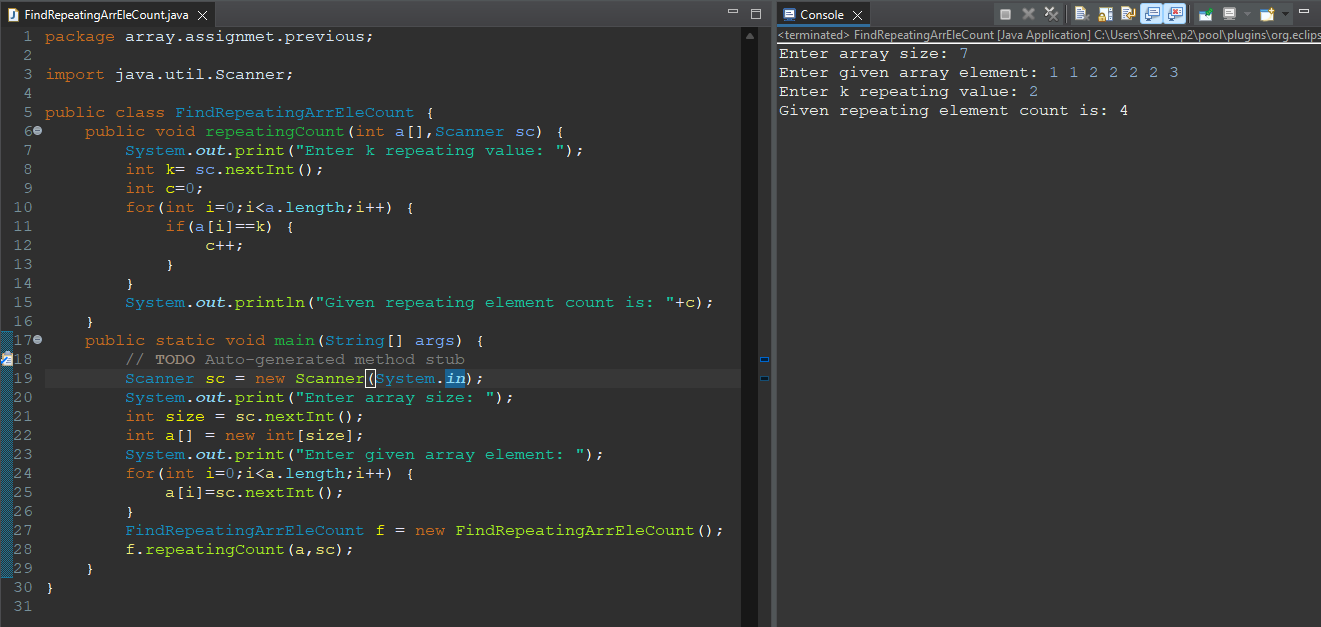
**Output: 7**

****

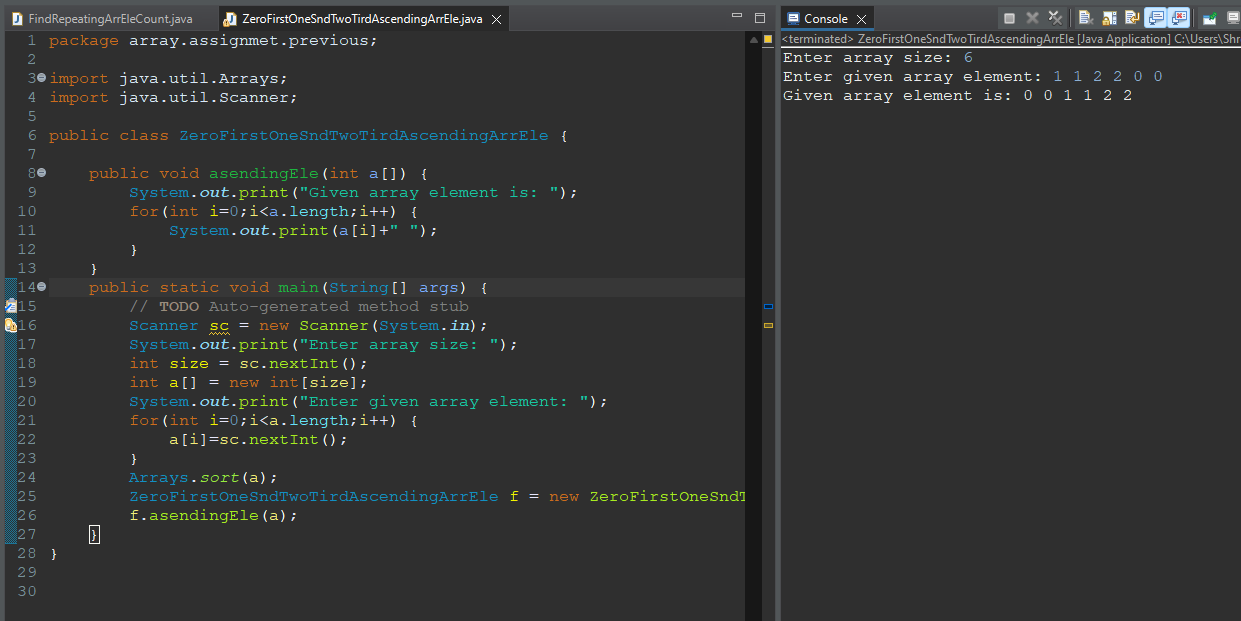
**Q3.Given a sorted array arr[] and a number x, write a function that counts the occurrences of x in arr[]. Expected time complexity is O(Logn)**

**Examples: Input: arr[] = {1, 1, 2, 2, 2, 2, 3,}, x = 2**

**Output: 4 // x (or 2) occurs 4 times in arr[]**

****

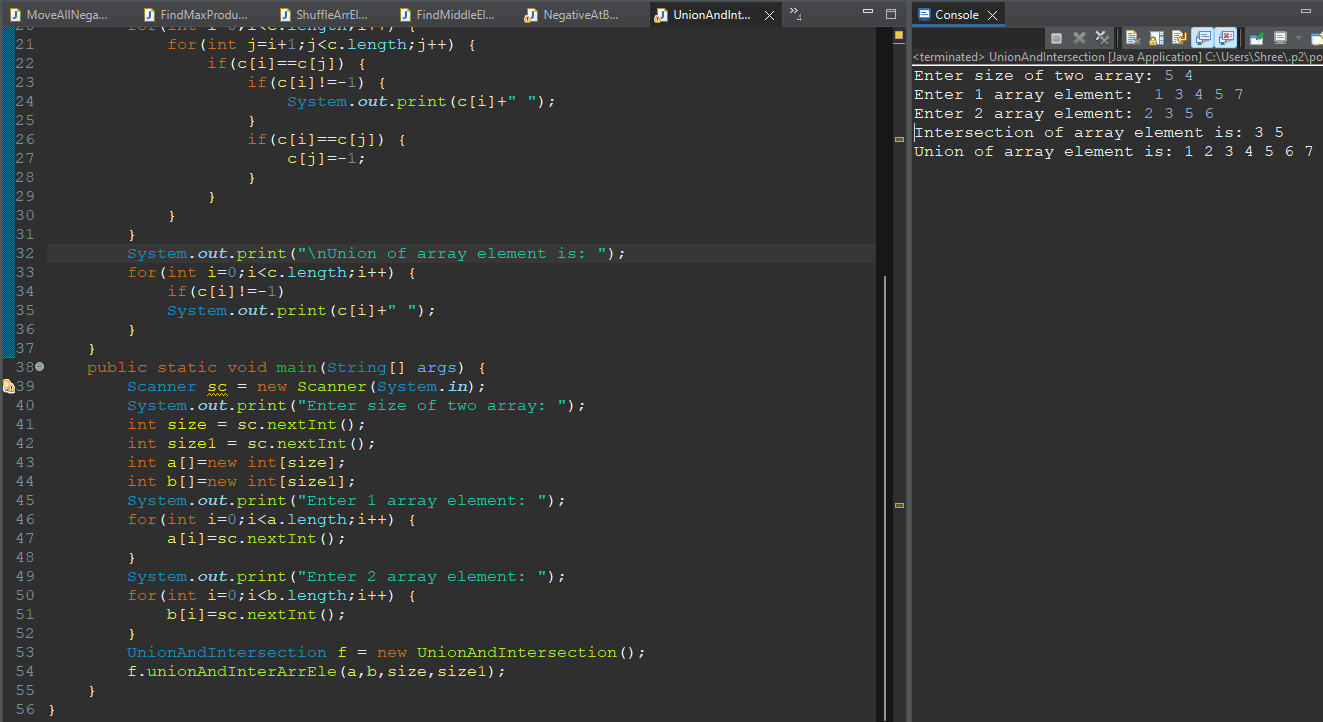
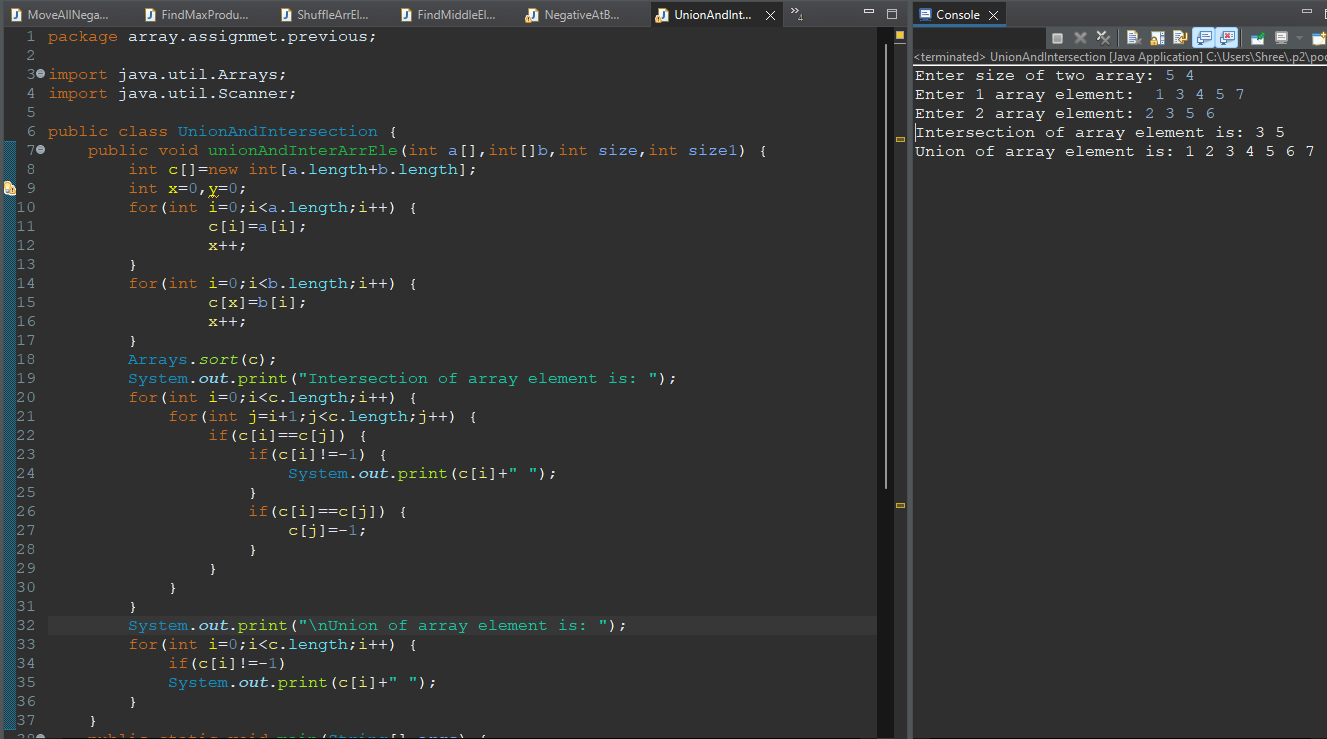
**Q4.Given an array A[] consisting of only 0s, 1s, and 2s. The task is to write a function that sorts the given array. The functions should put all 0s first, then all 1s and all 2s in last.**

****

**Q5.Given two sorted arrays, find their union and intersection.**

**Example: Input: arr1[] = {1, 3, 4, 5, 7} arr2[] = {2, 3, 5, 6}**

**Output: Union : {1, 2, 3, 4, 5, 6, 7} Intersection : {3, 5}**

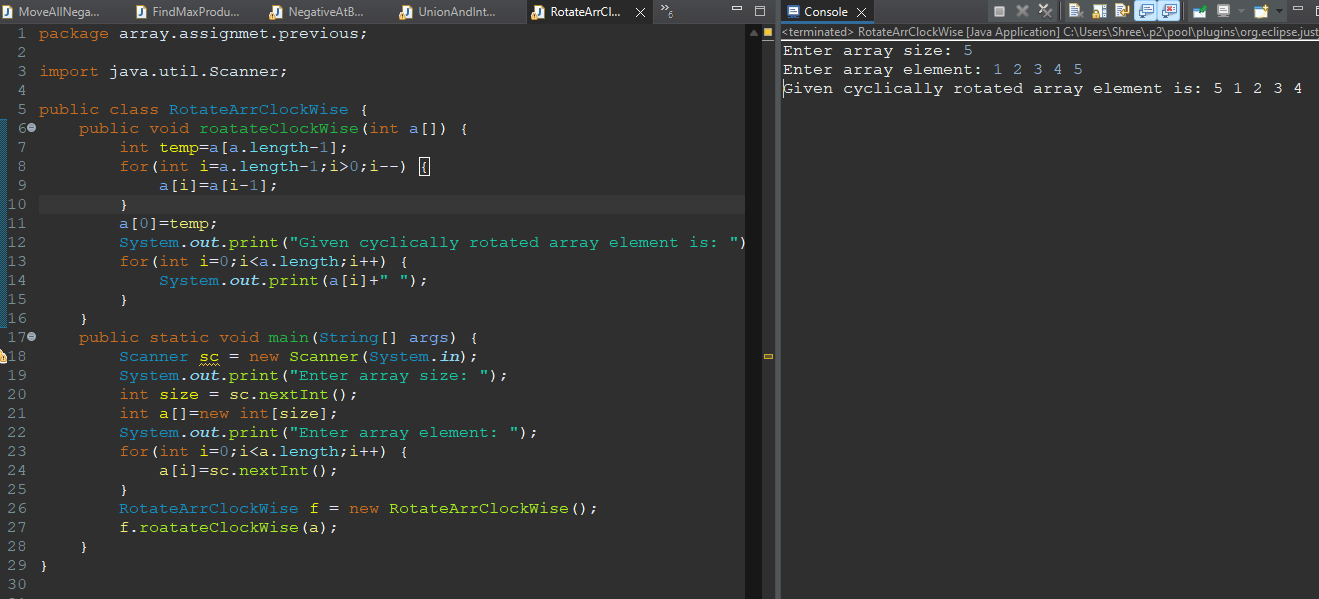
****

**Q6.Given an array, cyclically rotate the array clockwise by one.**

**Examples:**

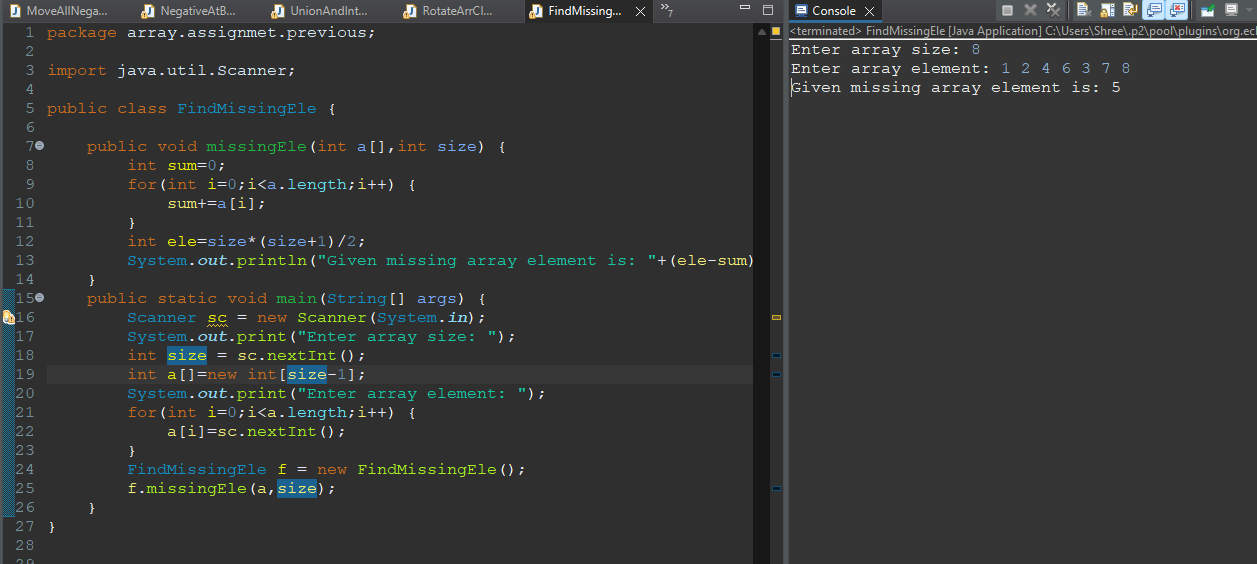
**Input: arr[] = {1, 2, 3, 4, 5}**

**Output: arr[] = {5, 1, 2, 3, 4}**

****

**Q7.Given an array arr[] of size N-1 with integers in the range of [1, N], the task is to find the missing number from the first N integers.Note: There are no duplicates in the list.**

**Examples: Input: arr[] = {1, 2, 4, 6, 3, 7, 8}, N = 8 Output: 5**

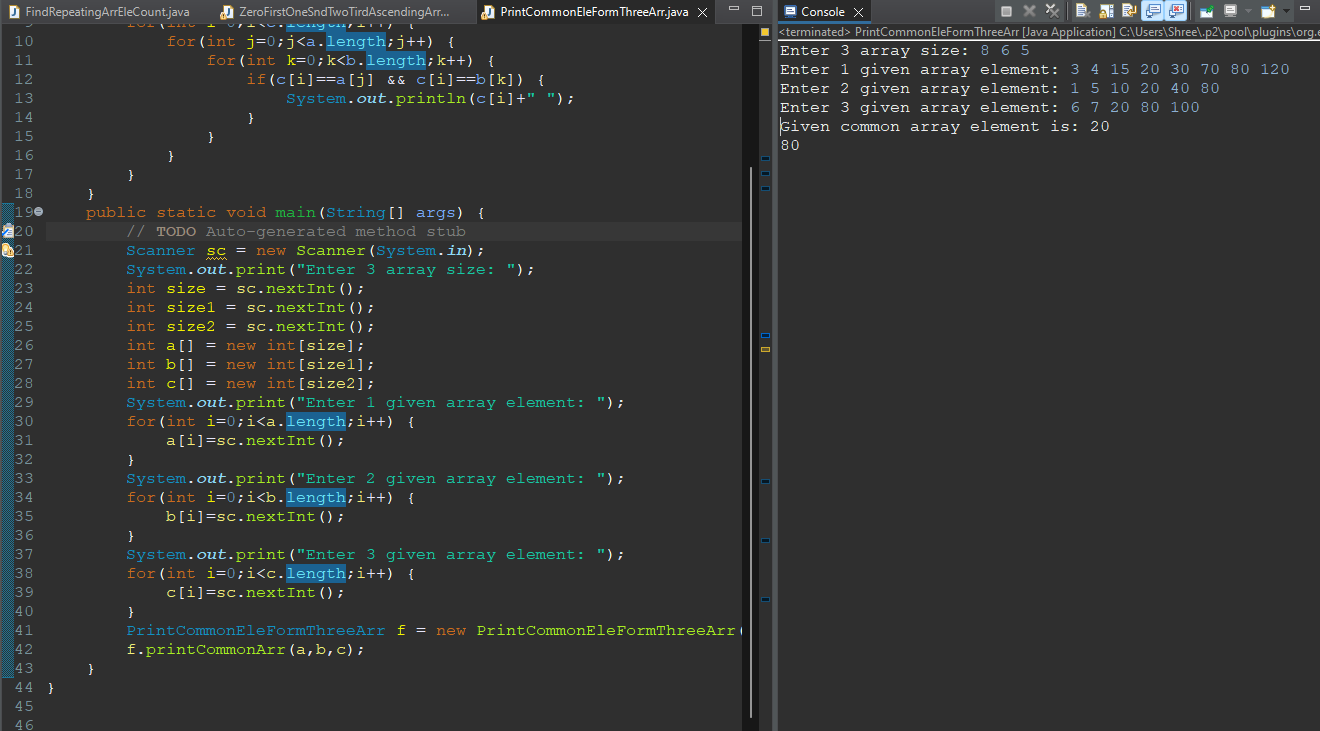
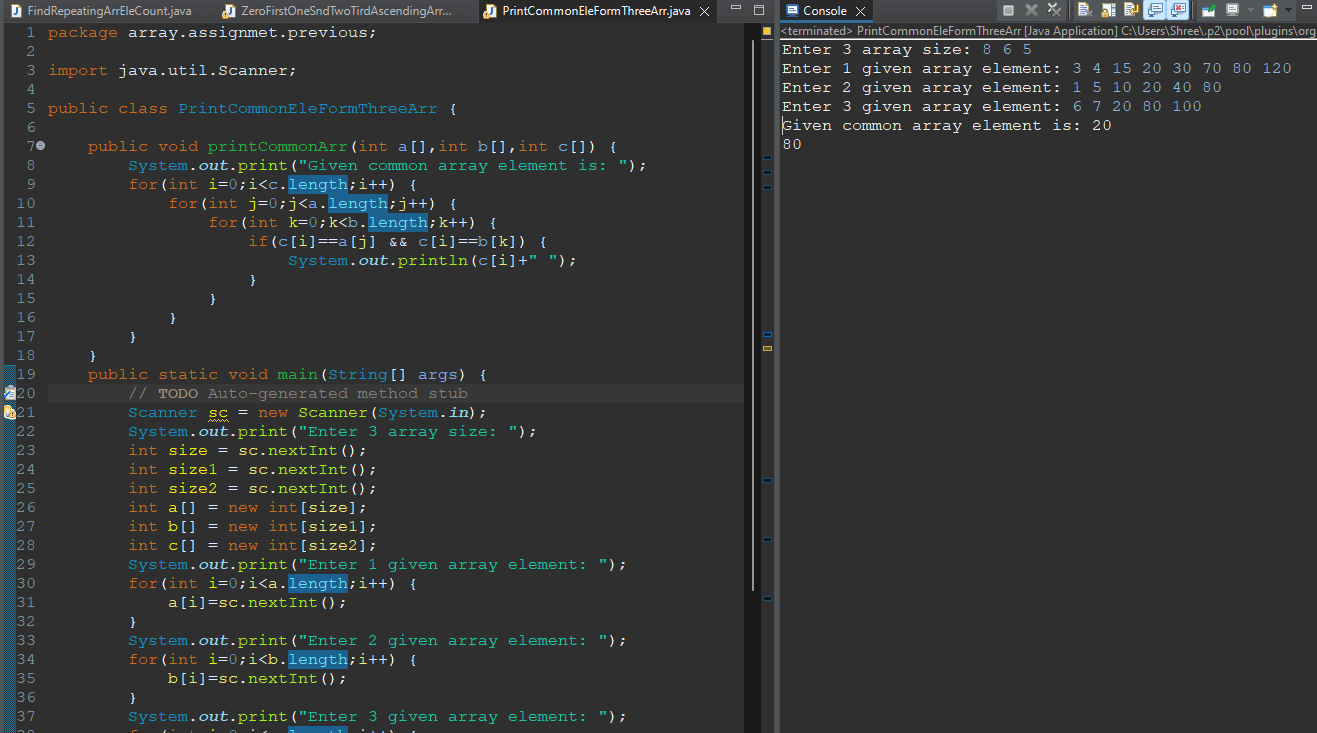
****

**Q8.Given three arrays sorted in non-decreasing order, print all common elements in these arrays.**

**Examples:**

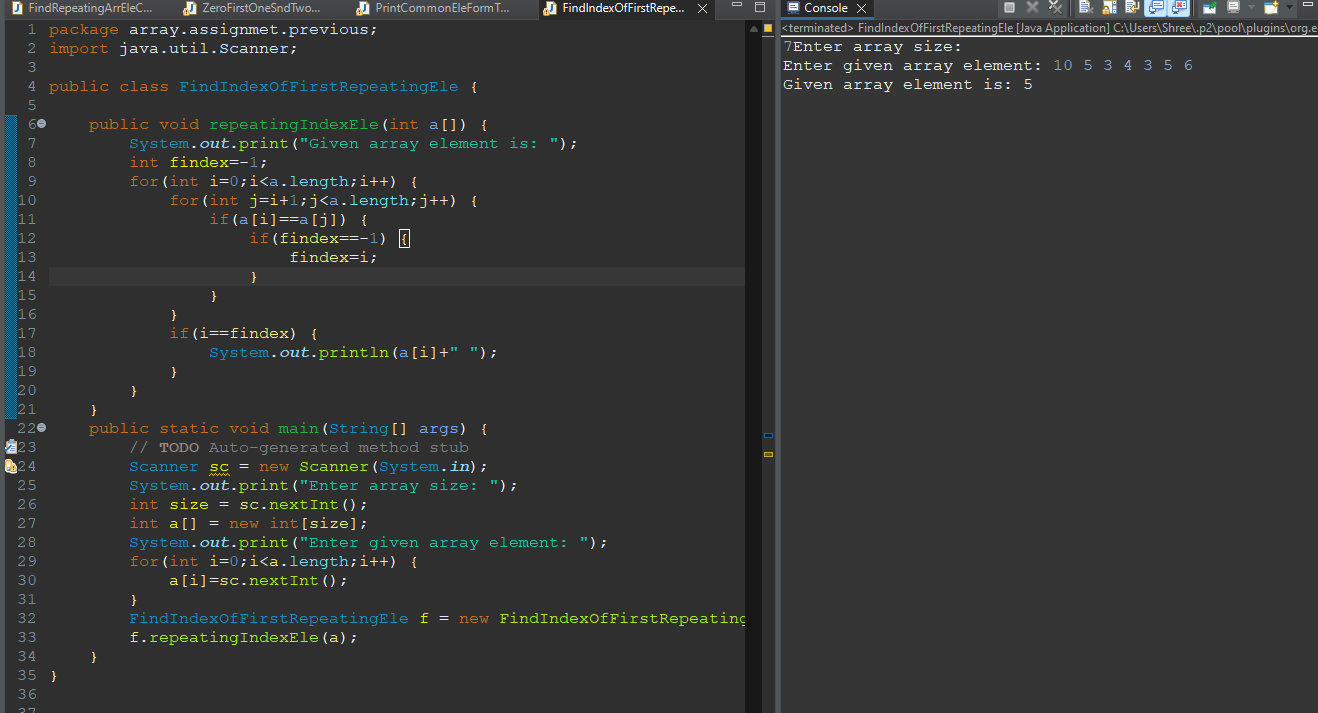
**Input: ar1[] = {1, 5, 10, 20, 40, 80} ar2[] = {6, 7, 20, 80, 100} ar3[] = {3, 4, 15, 20, 30, 70, 80, 120}**

**Output: 20, 80**

****

**Q9.Given an array of integers arr[], The task is to find the index of first repeating element in it i.e. the element that occurs more than once and whose index of the first occurrence is the smallest.**

**Examples: Input: arr[] = {10, 5, 3, 4, 3, 5, 6} Output: 5**

****

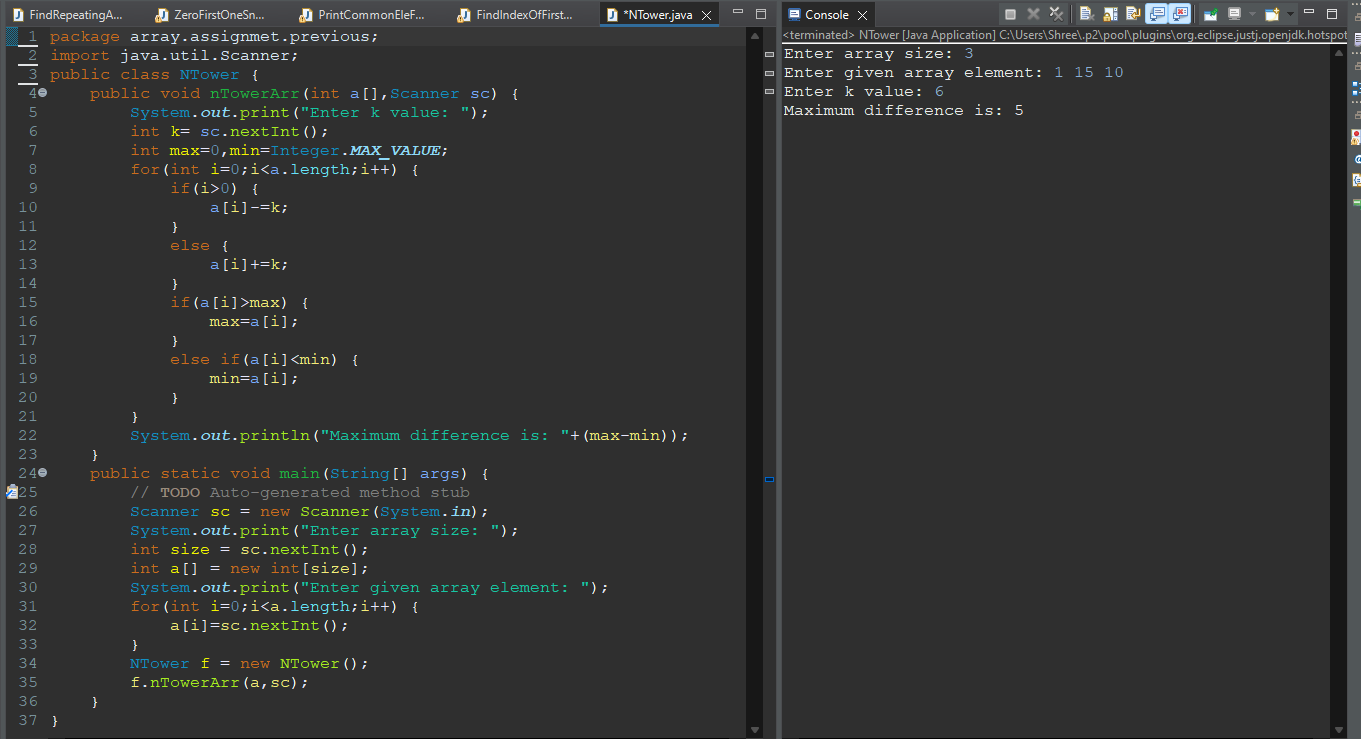
**Q10.Given the heights of N towers and a value of K, Either increase or decrease the height of every tower by K (only once) where K > 0. After modifications,**

**the task is to minimize the difference between the heights of the longest and the shortest tower and output its difference.**

**Examples: Input: arr[] = {1, 15, 10}, k = 6**

**Output: Maximum difference is 5.**

**Explanation: Change 1 to 7, 15 to 9 and 10 to 4. Maximum difference is 5 (between 4 and 9). We can’t get a lower difference.**

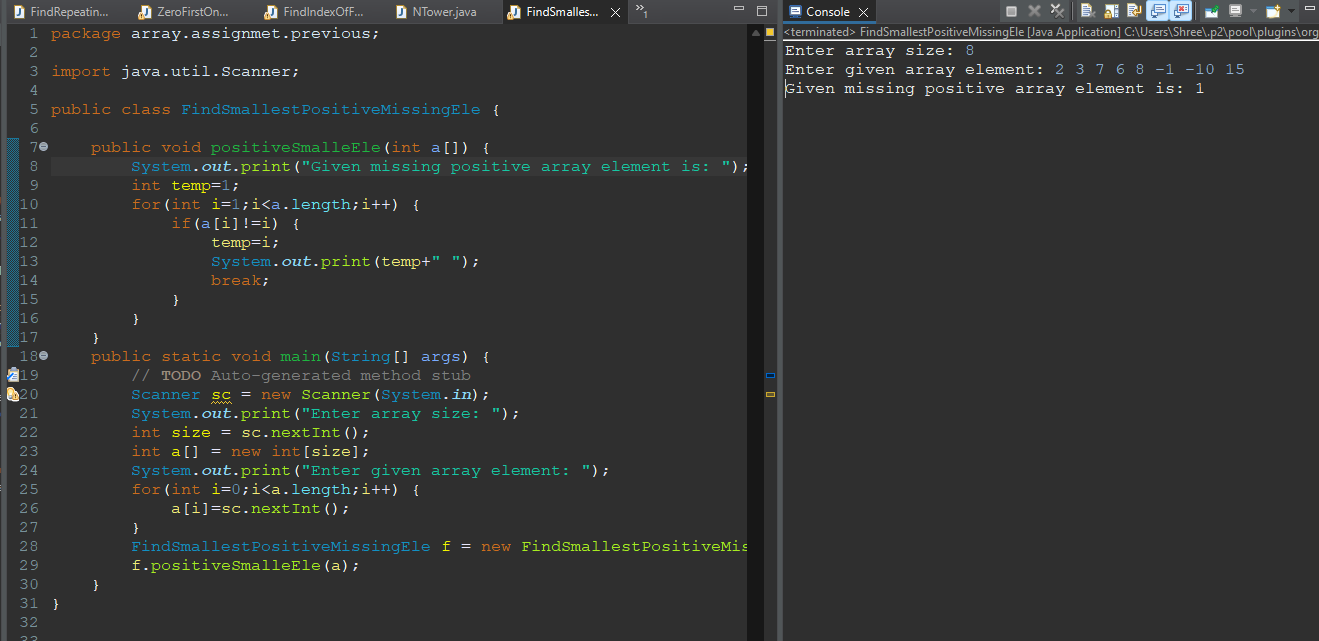
****

**Q11.Given an unsorted array arr[] with both positive and negative elements, the task is to find the smallest positive number missing from the array.**

**Note: You can modify the original array.**

**Examples: Input: arr[] = {2, 3, 7, 6, 8, -1, -10, 15}**

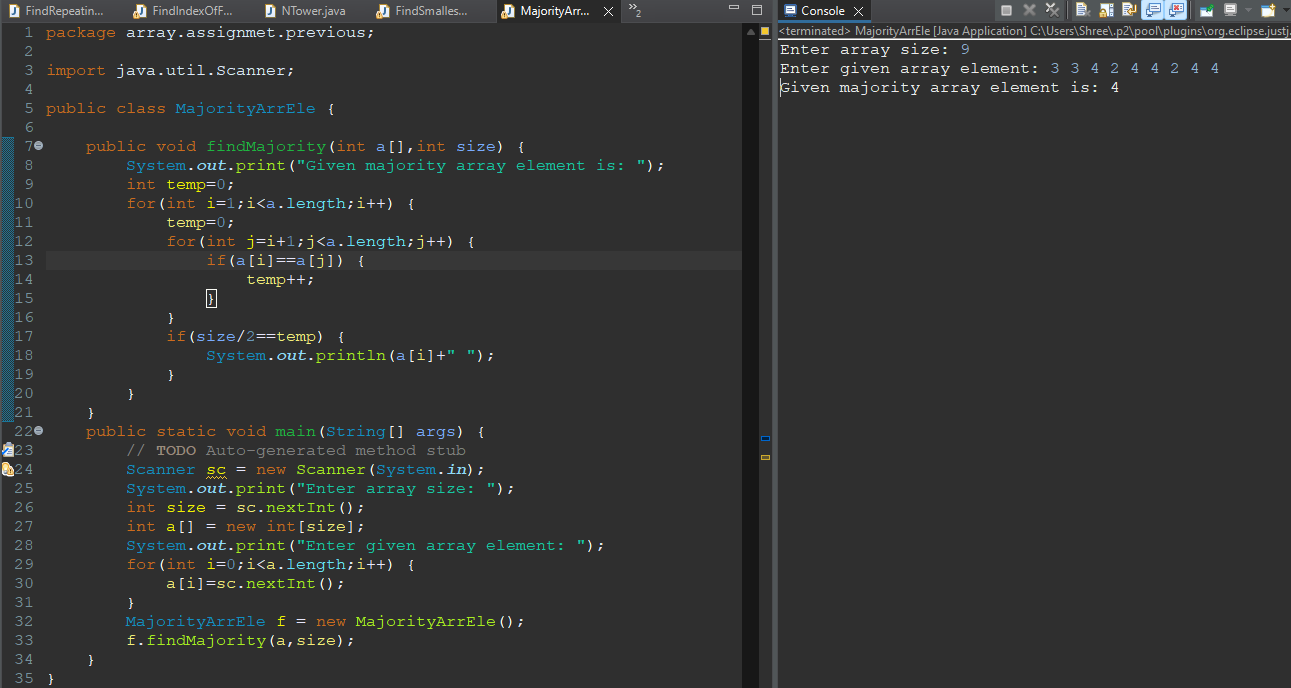
**Output: 1**

****

**Q12.Find the majority element in the array. A majority element in an array A[] of size n is an element that appears more than n/2 times (and hence there is at most one such element).**

**Examples: Input: {3, 3, 4, 2, 4, 4, 2, 4, 4}**

**Output: 4**

****

**Q13. Given two sorted arrays A and B of size p and q, write a Java program to merge elements of A with B by maintaining the sorted order i.e.**

**Fill A with first p smallest elements and fill B with remaining elements.**

**Example:**

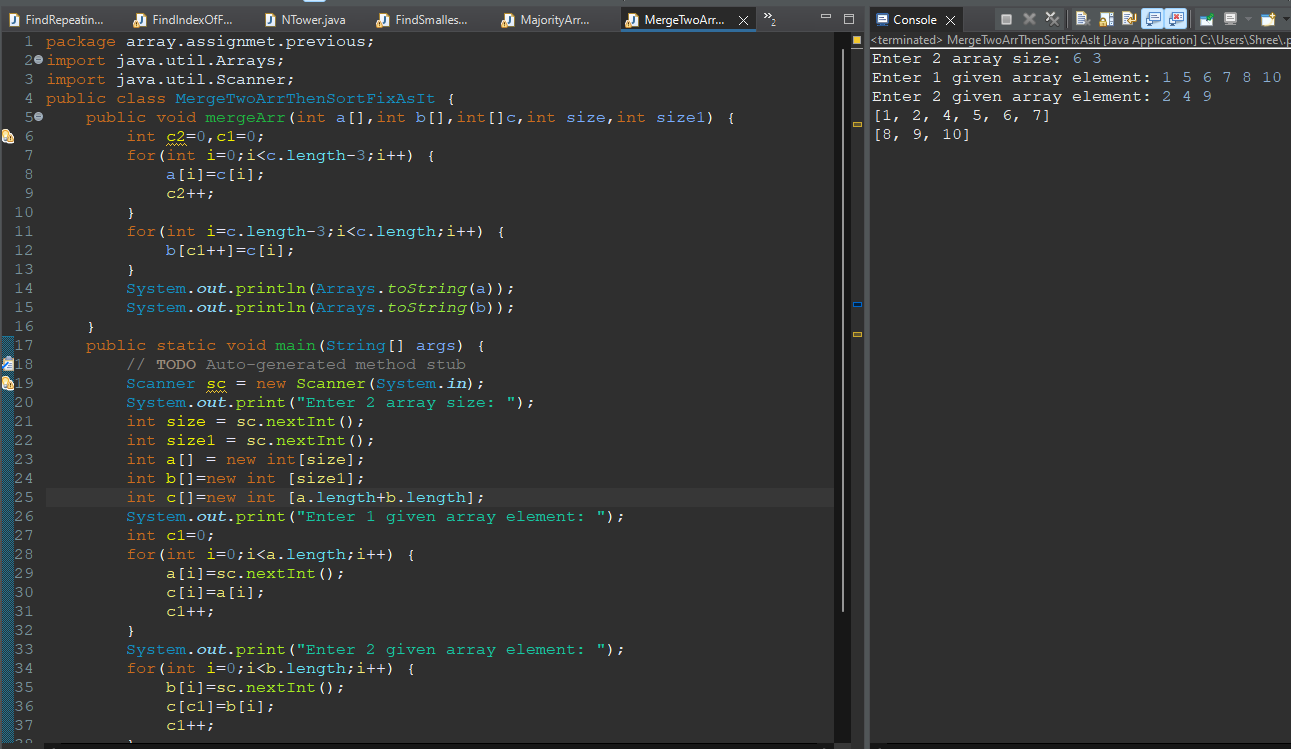
**Input: int[] A = { 1, 5, 6, 7, 8, 10 } int[] B = { 2, 4, 9 }**

**Output:**

**Sorted Arrays:**

**A: [1, 2, 4, 5, 6, 7]**

**B: [8, 9, 10]**

****

**Q14.Write a Java program to find maximum product of two integers in a given array of integers.**

**Example:**

**Input: nums = { 2, 3, 5, 7, -7, 5, 8, -5 }**

**Output:**

**Pair is (7, 8), Maximum Product: 56**

