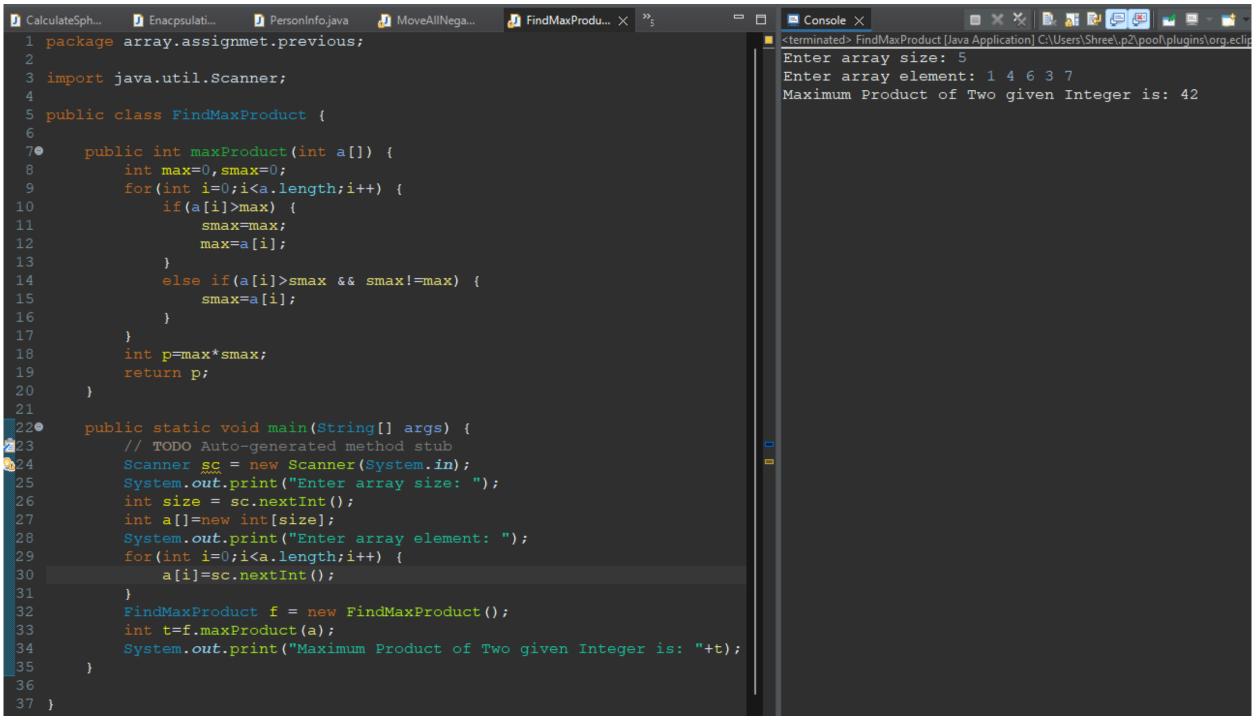
Assignment No:-28

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.



```java
package array.assignmet.previous;

import java.util.Scanner;

public class FindMaxProduct {

    public int maxProduct(int a[]) {
        int max=0,smax=0;
        for(int i=0;i<a.length;i++) {
            if(a[i]>max) {
                smax=max;
                max=a[i];
            }
            else if(a[i]>smax && smax!=max) {
                smax=a[i];
            }
        }
        int p=max*smax;
        return p;
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter array size: ");
        int size = sc.nextInt();
        int a[]=new int[size];
        System.out.print("Enter array element: ");
        for(int i=0;i<a.length;i++) {
            a[i]=sc.nextInt();
        }
        FindMaxProduct f = new FindMaxProduct();
        int t=f.maxProduct(a);
        System.out.print("Maximum Product of Two given Integer is: "+t);
    }

}
```

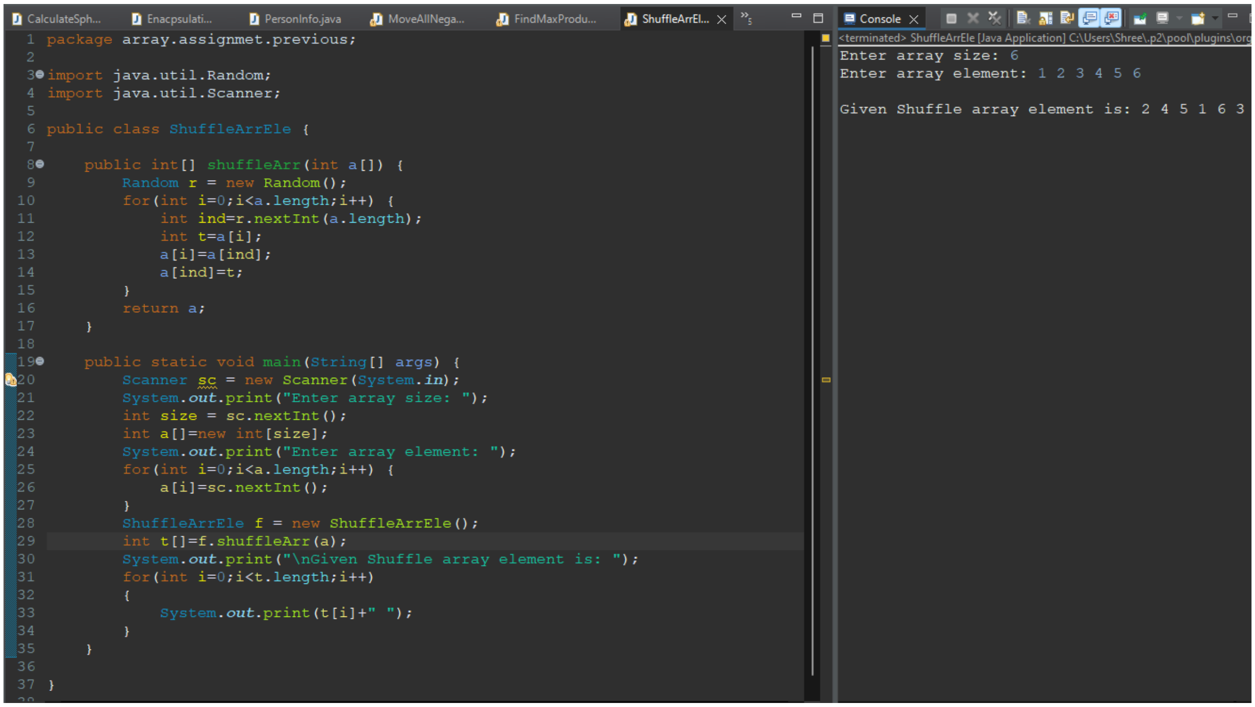
```
<terminated> FindMaxProduct [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclip
Enter array size: 5
Enter array element: 1 4 6 3 7
Maximum Product of Two given Integer is: 42
```

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]



```java
package array.assignmet.previous;

import java.util.Random;
import java.util.Scanner;

public class ShuffleArrEle {

    public int[] shuffleArr(int a[]) {
        Random r = new Random();
        for(int i=0;i<a.length;i++) {
            int ind=r.nextInt(a.length);
            int t=a[i];
            a[i]=a[ind];
            a[ind]=t;
        }
        return a;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter array size: ");
        int size = sc.nextInt();
        int a[]=new int[size];
        System.out.print("Enter array element: ");
        for(int i=0;i<a.length;i++) {
            a[i]=sc.nextInt();
        }
        ShuffleArrEle f = new ShuffleArrEle();
        int t[]=f.shuffleArr(a);
        System.out.print("\nGiven Shuffle array element is: ");
        for(int i=0;i<t.length;i++)
        {
            System.out.print(t[i]+" ");
        }
    }

}
```

```
<terminated> ShuffleArrEle [Java Application] C:\Users\Shree\.p2\pool\plugins\org
Enter array size: 6
Enter array element: 1 2 3 4 5 6

Given Shuffle array element is: 2 4 5 1 6 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:

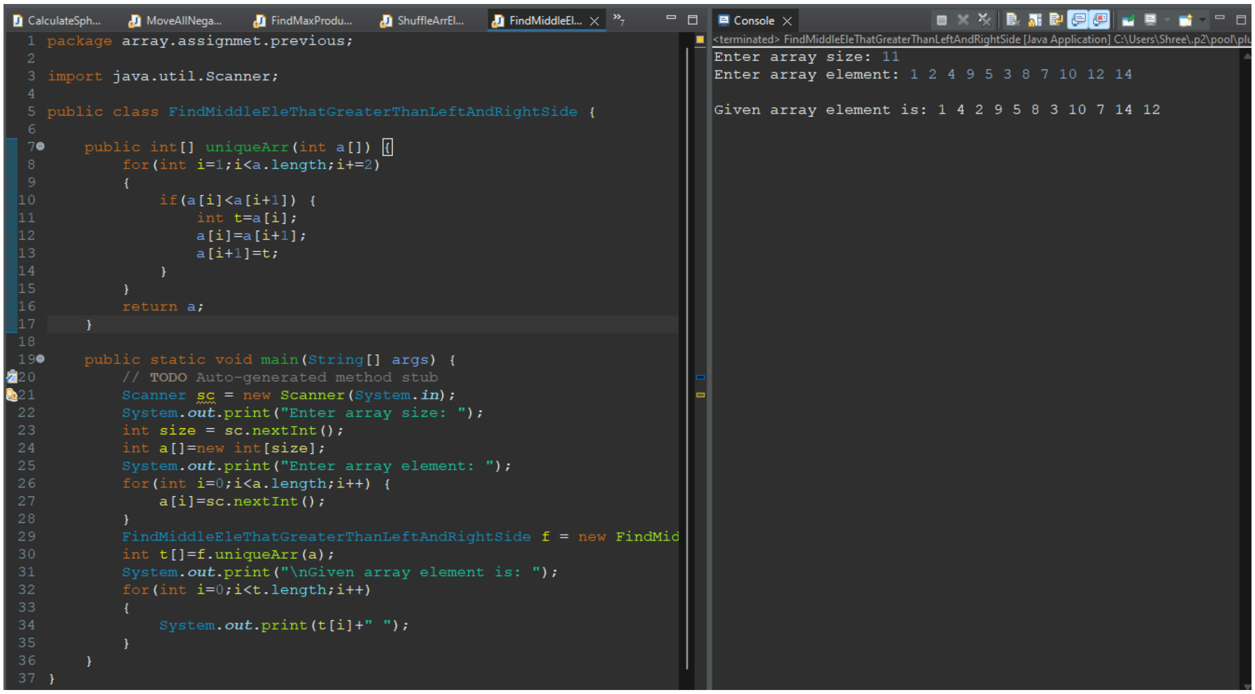
Input :

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]



```java
package array.assignmet.previous;

import java.util.Scanner;

public class FindMiddleEleThatGreaterThanLeftAndRightSide {

    public int[] uniqueArr(int a[]) {
        for(int i=1;i<a.length;i+=2)
        {
            if(a[i]<a[i+1]) {
                int t=a[i];
                a[i]=a[i+1];
                a[i+1]=t;
            }
        }
        return a;
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter array size: ");
        int size = sc.nextInt();
        int a[]=new int[size];
        System.out.print("Enter array element: ");
        for(int i=0;i<a.length;i++) {
            a[i]=sc.nextInt();
        }
        FindMiddleEleThatGreaterThanLeftAndRightSide f = new FindMid
        int t[]=f.uniqueArr(a);
        System.out.print("\nGiven array element is: ");
        for(int i=0;i<t.length;i++)
        {
            System.out.print(t[i]+" ");
        }
    }
}
```

```
<terminated> FindMiddleEleThatGreaterThanLeftAndRightSide [Java Application] C:\Users\Shree\.p2\pool\plu
Enter array size: 11
Enter array element: 1 2 4 9 5 3 8 7 10 12 14

Given array element is: 1 4 2 9 5 8 3 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.

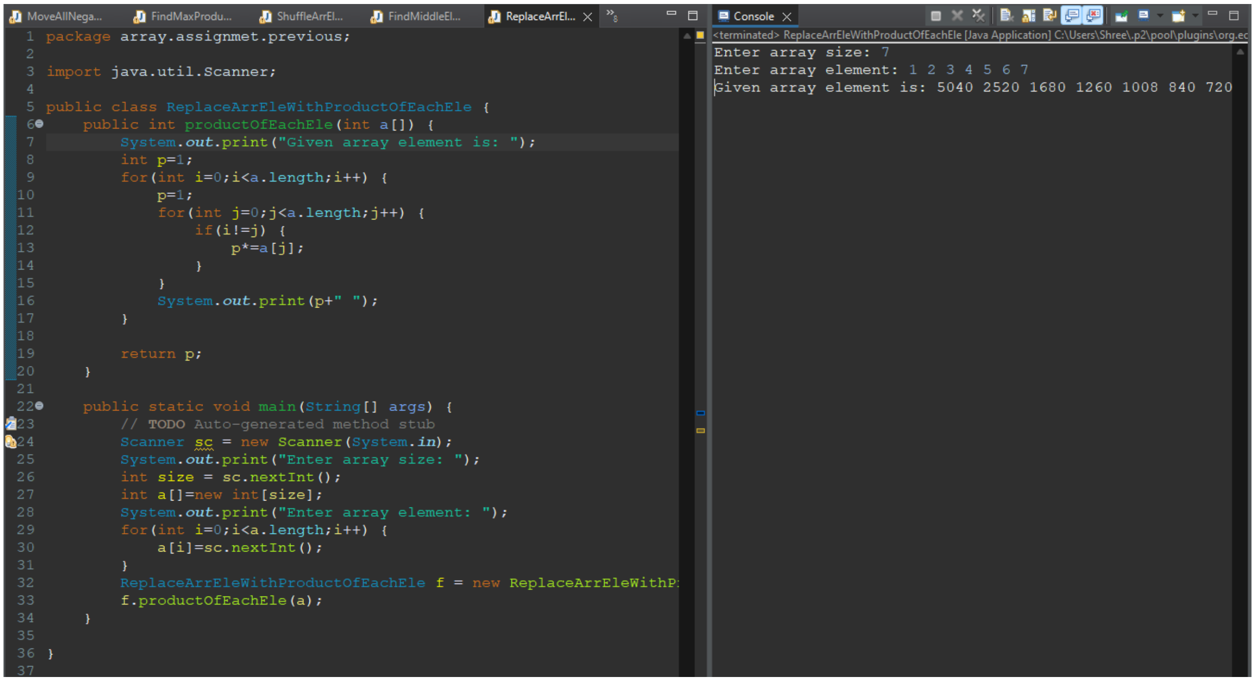
Example:

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

Output:

Array with product of every other element:

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



```java
package array.assignmet.previous;

import java.util.Scanner;

public class ReplaceArrEleWithProductOfEachEle {
    public int productOfEachEle(int a[]) {
        System.out.print("Given array element is: ");
        int p=1;
        for(int i=0;i<a.length;i++) {
            p=1;
            for(int j=0;j<a.length;j++) {
                if(i!=j) {
                    p*=a[j];
                }
            }
            System.out.print(p+" ");
        }

        return p;
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter array size: ");
        int size = sc.nextInt();
        int a[]=new int[size];
        System.out.print("Enter array element: ");
        for(int i=0;i<a.length;i++) {
            a[i]=sc.nextInt();
        }
        ReplaceArrEleWithProductOfEachEle f = new ReplaceArrEleWithP
        f.productOfEachEle(a);
    }

}
```

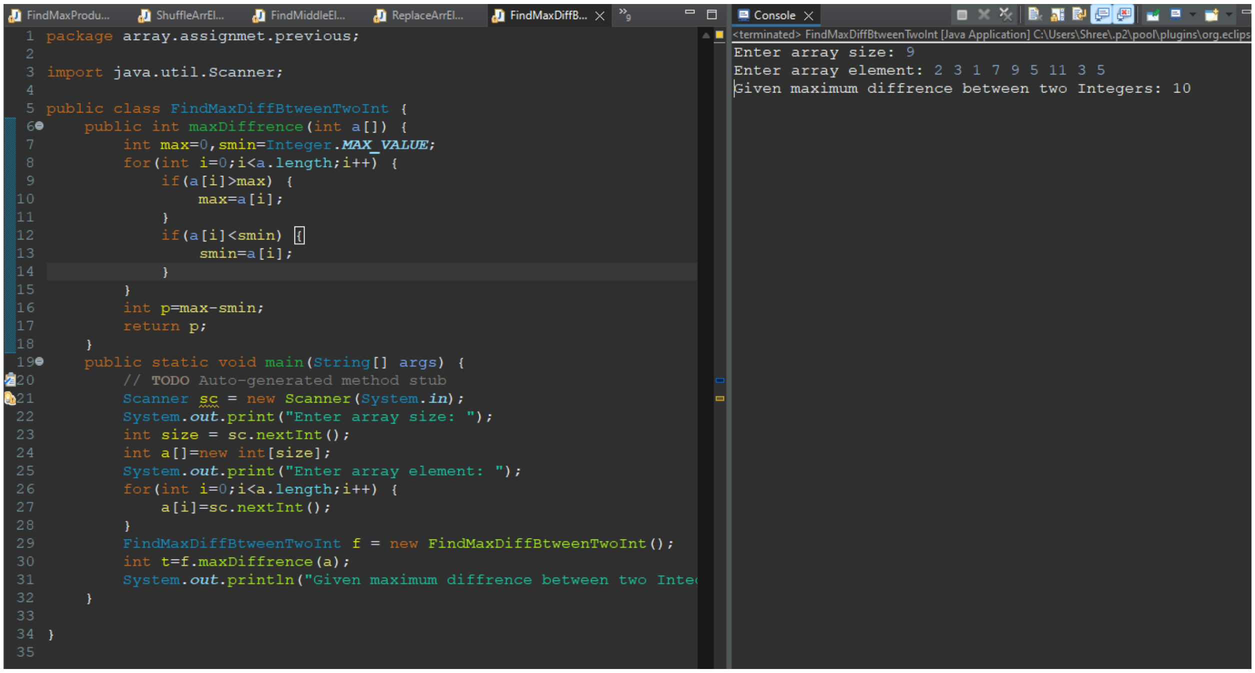
```
<terminated> ReplaceArrEleWithProductOfEachEle [Java Application] C:\Users\Shree\.p2\pool\plugins\org.ec
Enter array size: 7
Enter array element: 1 2 3 4 5 6 7
Given array element is: 5040 2520 1680 1260 1008 840 720
```

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



```java
package array.assignmet.previous;

import java.util.Scanner;

public class FindMaxDiffBtweenTwoInt {
    public int maxDiffrence(int a[]) {
        int max=0,smin=Integer.MAX_VALUE;
        for(int i=0;i<a.length;i++) {
            if(a[i]>max) {
                max=a[i];
            }
            if(a[i]<smin) {
                smin=a[i];
            }
        }
        int p=max-smin;
        return p;
    }
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter array size: ");
        int size = sc.nextInt();
        int a[]=new int[size];
        System.out.print("Enter array element: ");
        for(int i=0;i<a.length;i++) {
            a[i]=sc.nextInt();
        }
        FindMaxDiffBtweenTwoInt f = new FindMaxDiffBtweenTwoInt();
        int t=f.maxDiffrence(a);
        System.out.println("Given maximum diffrence between two Inte
    }

}
```

```
<terminated> FindMaxDiffBtweenTwoInt [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclips
Enter array size: 9
Enter array element: 2 3 1 7 9 5 11 3 5
Given maximum diffrence between two Integers: 10
```

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

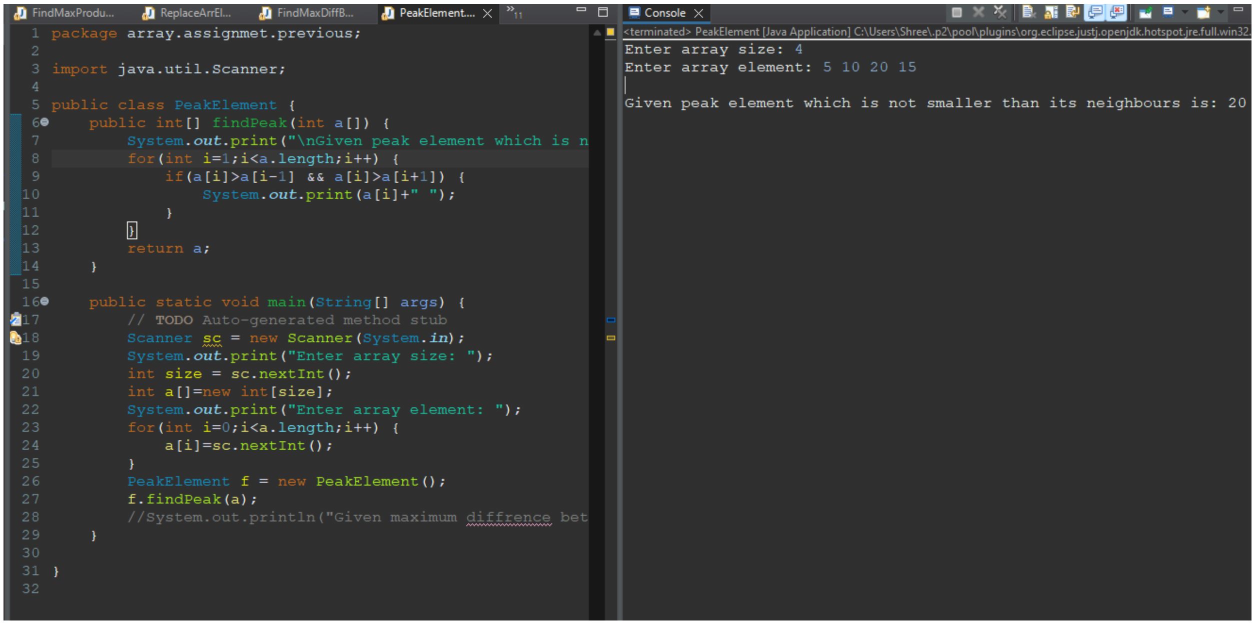
Input: array[]= {5, 10, 20, 15} Output: 20

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}

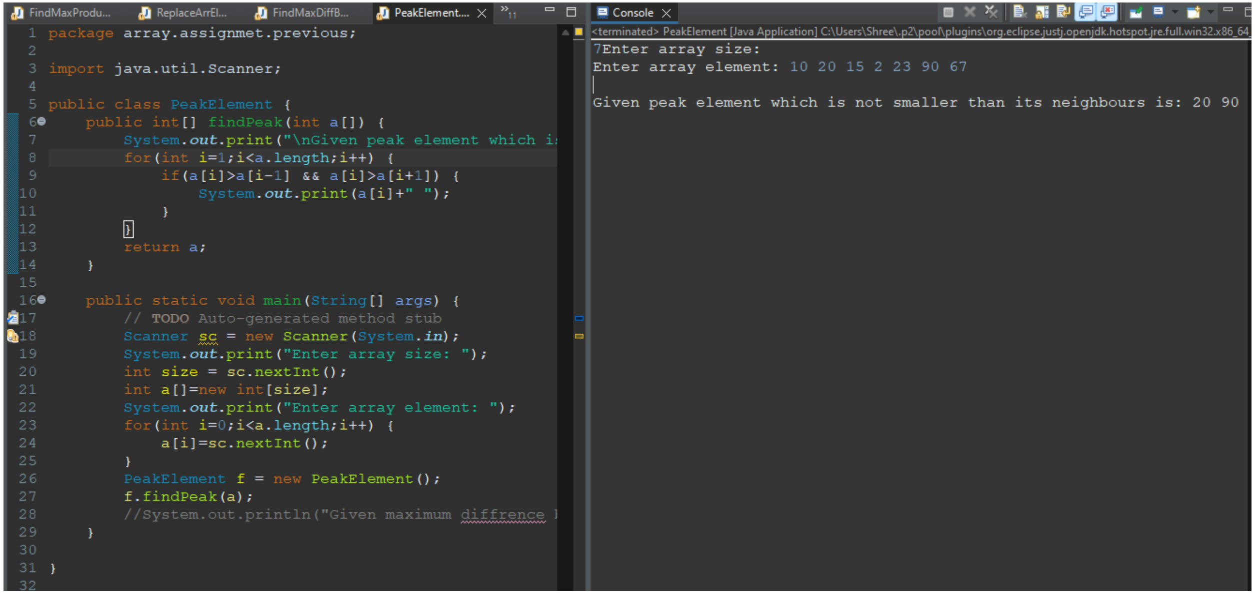
Output: 20 or 90

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



```java
package array.assignmet.previous;

import java.util.Scanner;

public class PeakElement {
    public int[] findPeak(int a[]) {
        System.out.print("\nGiven peak element which is n
        for(int i=1;i<a.length;i++) {
            if(a[i]>a[i-1] && a[i]>a[i+1]) {
                System.out.print(a[i]+" ");
            }
        }
        return a;
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter array size: ");
        int size = sc.nextInt();
        int a[]=new int[size];
        System.out.print("Enter array element: ");
        for(int i=0;i<a.length;i++) {
            a[i]=sc.nextInt();
        }
        PeakElement f = new PeakElement();
        f.findPeak(a);
        //System.out.println("Given maximum diffrence bet
    }

}
```

```
<terminated> PeakElement [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.
Enter array size: 4
Enter array element: 5 10 20 15

Given peak element which is not smaller than its neighbours is: 20
```



```java
package array.assignmet.previous;

import java.util.Scanner;

public class PeakElement {
    public int[] findPeak(int a[]) {
        System.out.print("\nGiven peak element which i
        for(int i=1;i<a.length;i++) {
            if(a[i]>a[i-1] && a[i]>a[i+1]) {
                System.out.print(a[i]+" ");
            }
        }
        return a;
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter array size: ");
        int size = sc.nextInt();
        int a[]=new int[size];
        System.out.print("Enter array element: ");
        for(int i=0;i<a.length;i++) {
            a[i]=sc.nextInt();
        }
        PeakElement f = new PeakElement();
        f.findPeak(a);
        //System.out.println("Given maximum diffrence 
    }

}
```

```
<terminated> PeakElement [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64
7Enter array size: 
Enter array element: 10 20 15 2 23 90 67

Given peak element which is not smaller than its neighbours is: 20 90
```

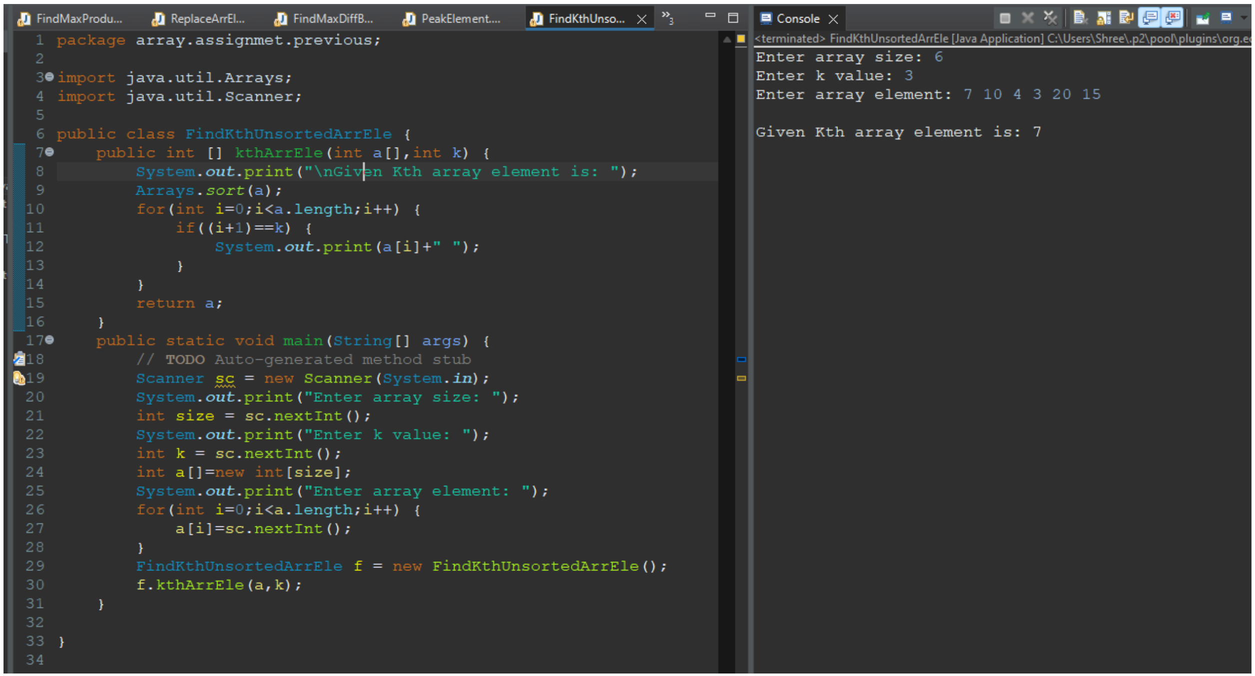
7.K’th Largest Element in Unsorted Array

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

Output: 7

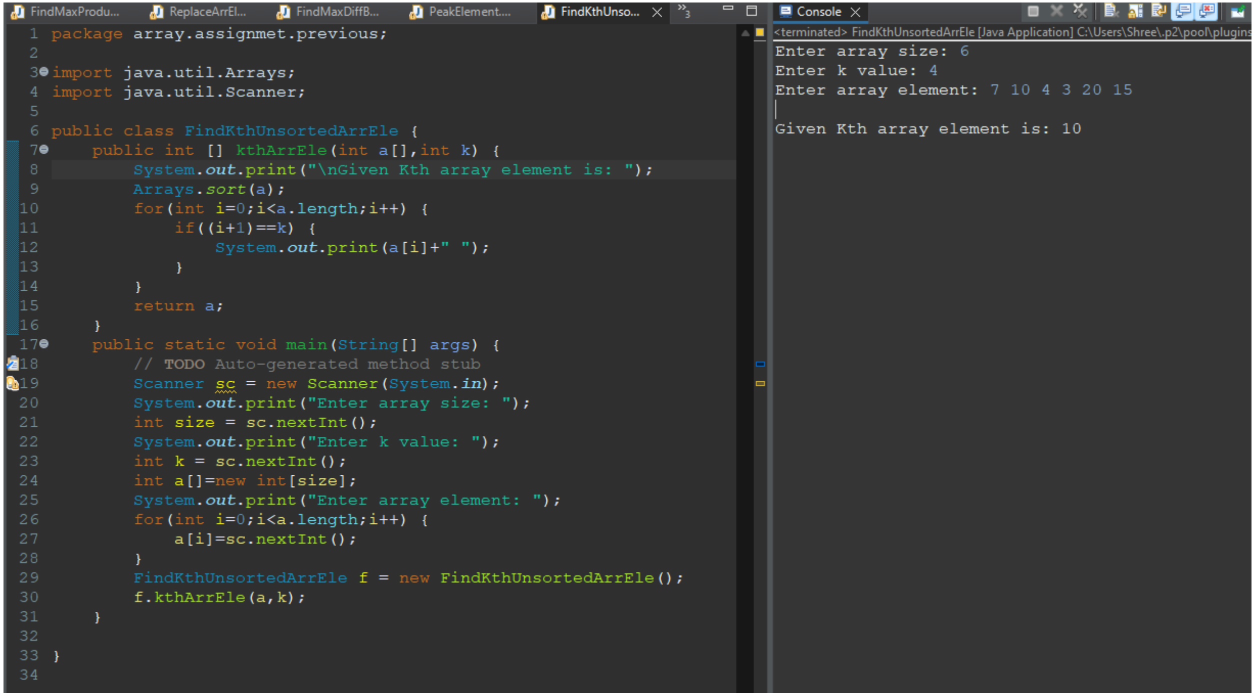
Input: arr[] = {7, 10, 4, 3, 20, 15}, K = 4

Output: 10



```java
package array.assignmet.previous;

import java.util.Arrays;
import java.util.Scanner;

public class FindKthUnsortedArrEle {
    public int [] kthArrEle(int a[],int k) {
        System.out.print("\nGiven Kth array element is: ");
        Arrays.sort(a);
        for(int i=0;i<a.length;i++) {
            if((i+1)==k) {
                System.out.print(a[i]+" ");
            }
        }
        return a;
    }
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter array size: ");
        int size = sc.nextInt();
        System.out.print("Enter k value: ");
        int k = sc.nextInt();
        int a[]=new int[size];
        System.out.print("Enter array element: ");
        for(int i=0;i<a.length;i++) {
            a[i]=sc.nextInt();
        }
        FindKthUnsortedArrEle f = new FindKthUnsortedArrEle();
        f.kthArrEle(a,k);
    }

}
```

```
<terminated> FindKthUnsortedArrEle [Java Application] C:\Users\Shree\.p2\pool\plugins\org.ec
Enter array size: 6
Enter k value: 3
Enter array element: 7 10 4 3 20 15

Given Kth array element is: 7
```



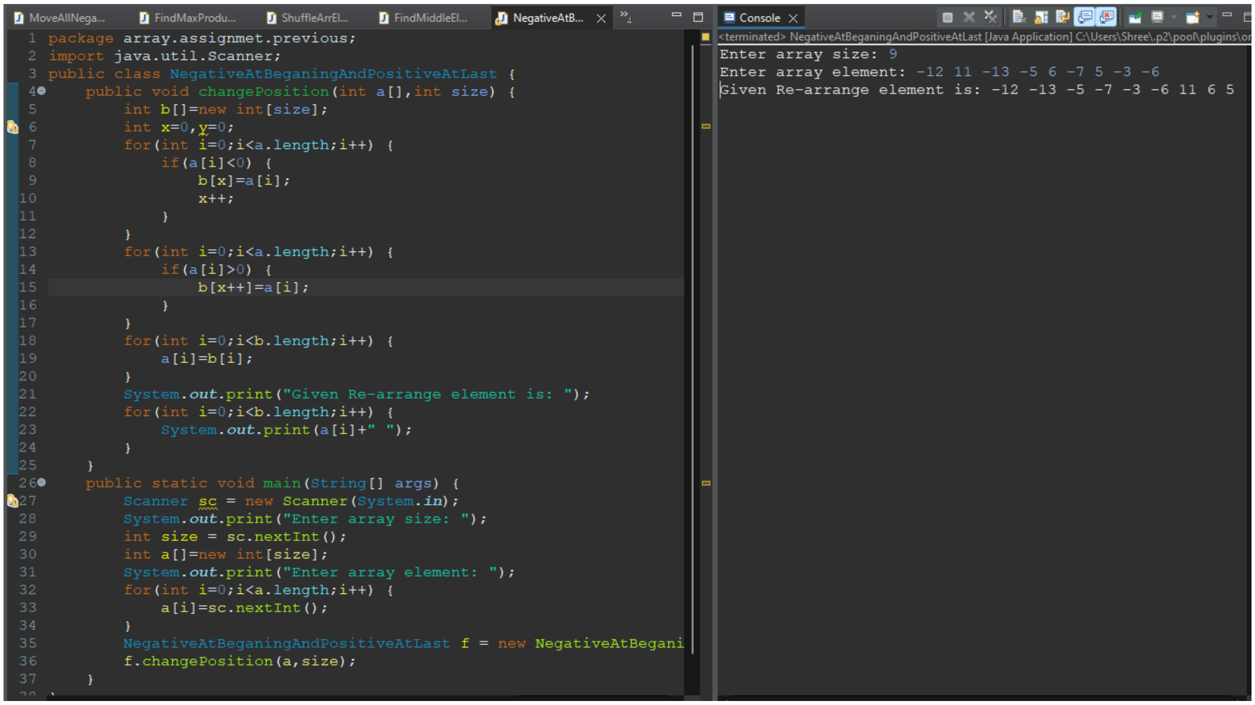
```java
package array.assignmet.previous;

import java.util.Arrays;
import java.util.Scanner;

public class FindKthUnsortedArrEle {
    public int [] kthArrEle(int a[],int k) {
        System.out.print("\nGiven Kth array element is: ");
        Arrays.sort(a);
        for(int i=0;i<a.length;i++) {
            if((i+1)==k) {
                System.out.print(a[i]+" ");
            }
        }
        return a;
    }
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter array size: ");
        int size = sc.nextInt();
        System.out.print("Enter k value: ");
        int k = sc.nextInt();
        int a[]=new int[size];
        System.out.print("Enter array element: ");
        for(int i=0;i<a.length;i++) {
            a[i]=sc.nextInt();
        }
        FindKthUnsortedArrEle f = new FindKthUnsortedArrEle();
        f.kthArrEle(a,k);
    }

}
```

```
<terminated> FindKthUnsortedArrEle [Java Application] C:\Users\Shree\.p2\pool\plugins
Enter array size: 6
Enter k value: 4
Enter array element: 7 10 4 3 20 15

Given Kth array element is: 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



```java
package array.assignmet.previous;
import java.util.Scanner;
public class NegativeAtBeganingAndPositiveAtLast {
    public void changePosition(int a[],int size) {
        int b[]=new int[size];
        int x=0,y=0;
        for(int i=0;i<a.length;i++) {
            if(a[i]<0) {
                b[x]=a[i];
                x++;
            }
        }
        for(int i=0;i<a.length;i++) {
            if(a[i]>0) {
                b[x++]=a[i];
            }
        }
        for(int i=0;i<b.length;i++) {
            a[i]=b[i];
        }
        System.out.print("Given Re-arrange element is: ");
        for(int i=0;i<b.length;i++) {
            System.out.print(a[i]+" ");
        }
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter array size: ");
        int size = sc.nextInt();
        int a[]=new int[size];
        System.out.print("Enter array element: ");
        for(int i=0;i<a.length;i++) {
            a[i]=sc.nextInt();
        }
        NegativeAtBeganingAndPositiveAtLast f = new NegativeAtBegani
        f.changePosition(a,size);
    }
```

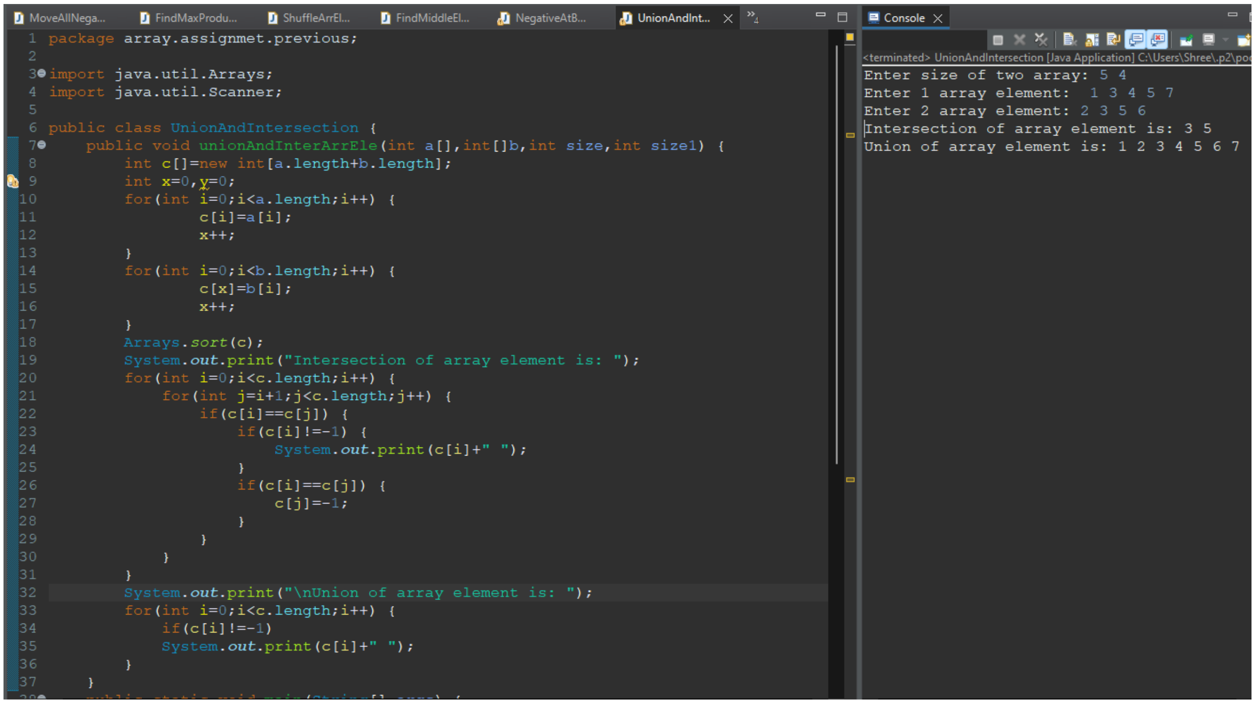
```
<terminated> NegativeAtBeganingAndPositiveAtLast [Java Application] C:\Users\Shree\.p2\pool\plugins\or
Enter array size: 9
Enter array element: -12 11 -13 -5 6 -7 5 -3 -6
Given Re-arrange element is: -12 -13 -5 -7 -3 -6 11 6 5
```

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}

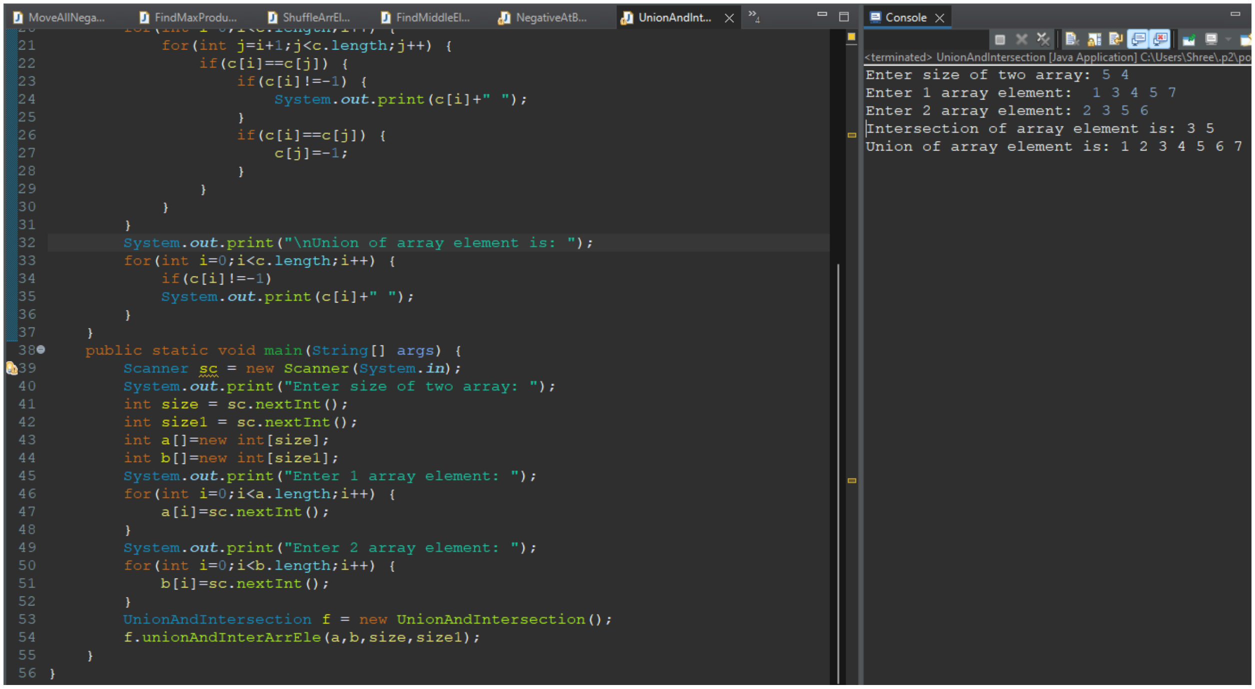
Input: arr1[] = {2, 5, 6}arr2[] = {4, 6, 8, 10}

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



```
package array.assignmet.previous;

import java.util.Arrays;
import java.util.Scanner;

public class UnionAndIntersection {
    public void unionAndInterArrEle(int a[],int[]b,int size,int size1) {
        int c[]=new int[a.length+b.length];
        int x=0,y=0;
        for(int i=0;i<a.length;i++) {
                c[i]=a[i];
                x++;
        }
        for(int i=0;i<b.length;i++) {
                c[x]=b[i];
                x++;
        }
        Arrays.sort(c);
        System.out.print("Intersection of array element is: ");
        for(int i=0;i<c.length;i++) {
            for(int j=i+1;j<c.length;j++) {
                if(c[i]==c[j]) {
                    if(c[i]!=-1) {
                        System.out.print(c[i]+" ");
                    }
                    if(c[i]==c[j]) {
                        c[j]=-1;
                    }
                }
            }
        }
        System.out.print("\nUnion of array element is: ");
        for(int i=0;i<c.length;i++) {
            if(c[i]!=-1)
            System.out.print(c[i]+" ");
        }
    }
```

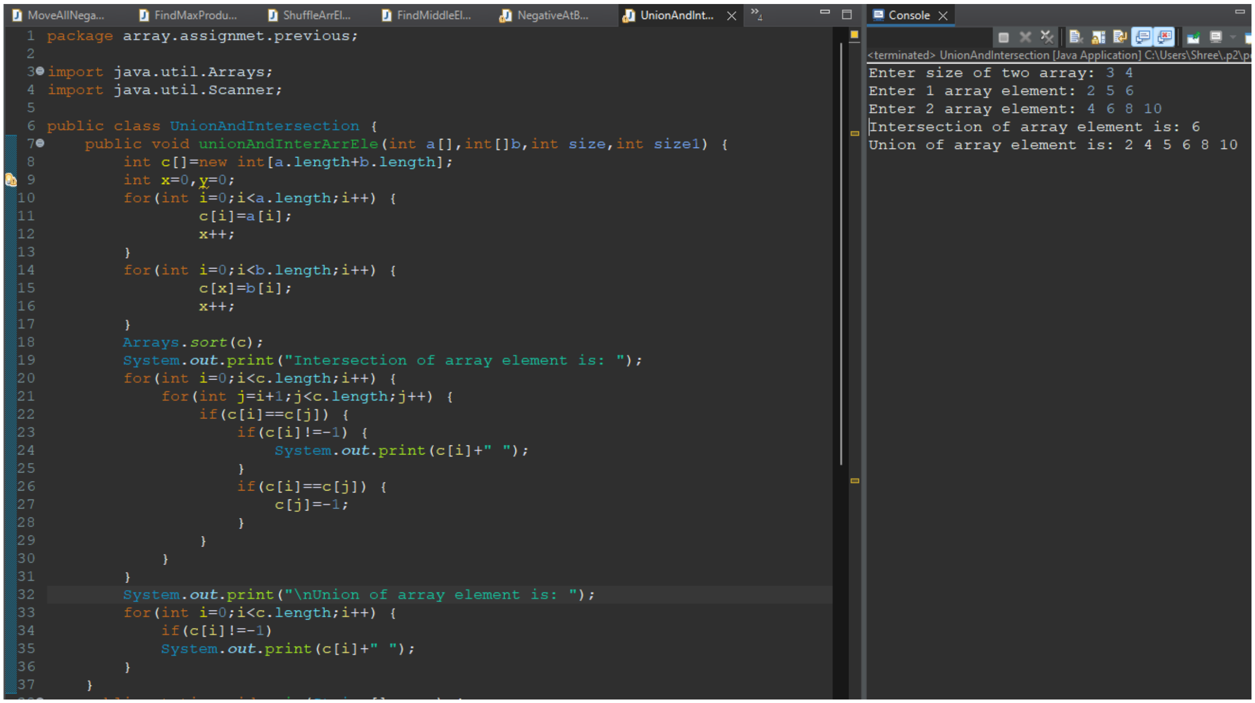
```
<terminated> UnionAndIntersection [Java Application] C:\Users\Shree\.p2\po
Enter size of two array: 5 4
Enter 1 array element:  1 3 4 5 7
Enter 2 array element: 2 3 5 6
Intersection of array element is: 3 5
Union of array element is: 1 2 3 4 5 6 7
```



```
            for(int j=i+1;j<c.length;j++) {
                if(c[i]==c[j]) {
                    if(c[i]!=-1) {
                        System.out.print(c[i]+" ");
                    }
                    if(c[i]==c[j]) {
                        c[j]=-1;
                    }
                }
            }
        }
        System.out.print("\nUnion of array element is: ");
        for(int i=0;i<c.length;i++) {
            if(c[i]!=-1)
            System.out.print(c[i]+" ");
        }
    }
```

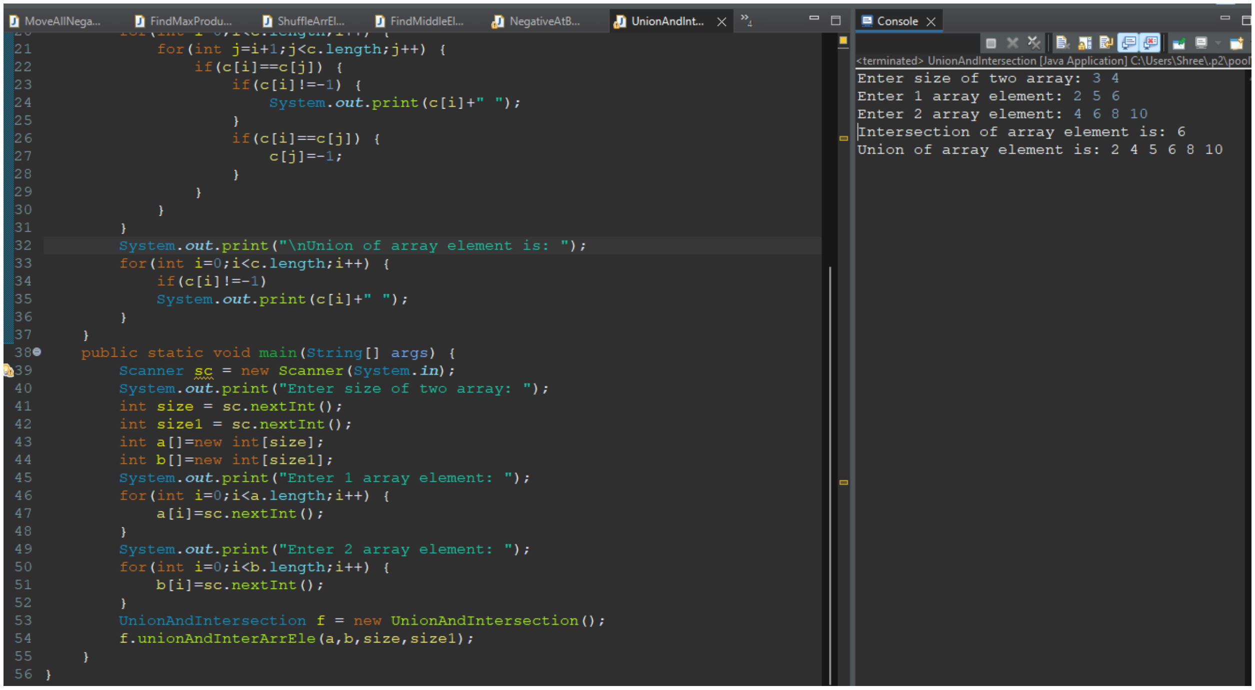
```
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter size of two array: ");
        int size = sc.nextInt();
        int size1 = sc.nextInt();
        int a[]=new int[size];
        int b[]=new int[size1];
        System.out.print("Enter 1 array element: ");
        for(int i=0;i<a.length;i++) {
            a[i]=sc.nextInt();
        }
        System.out.print("Enter 2 array element: ");
        for(int i=0;i<b.length;i++) {
            b[i]=sc.nextInt();
        }
        UnionAndIntersection f = new UnionAndIntersection();
        f.unionAndInterArrEle(a,b,size,size1);
    }
}
```

```
<terminated> UnionAndIntersection [Java Application] C:\Users\Shree\.p2\po
Enter size of two array: 5 4
Enter 1 array element:  1 3 4 5 7
Enter 2 array element: 2 3 5 6
Intersection of array element is: 3 5
Union of array element is: 1 2 3 4 5 6 7
```



```
package array.assignmet.previous;

import java.util.Arrays;
import java.util.Scanner;

public class UnionAndIntersection {
    public void unionAndInterArrEle(int a[],int[]b,int size,int size1) {
        int c[]=new int[a.length+b.length];
        int x=0,y=0;
        for(int i=0;i<a.length;i++) {
                c[i]=a[i];
                x++;
        }
        for(int i=0;i<b.length;i++) {
                c[x]=b[i];
                x++;
        }
        Arrays.sort(c);
        System.out.print("Intersection of array element is: ");
        for(int i=0;i<c.length;i++) {
            for(int j=i+1;j<c.length;j++) {
                if(c[i]==c[j]) {
                    if(c[i]!=-1) {
                        System.out.print(c[i]+" ");
                    }
                    if(c[i]==c[j]) {
                        c[j]=-1;
                    }
                }
            }
        }
        System.out.print("\nUnion of array element is: ");
        for(int i=0;i<c.length;i++) {
            if(c[i]!=-1)
            System.out.print(c[i]+" ");
        }
    }
    public static void main(String[] args) {
```

```
<terminated> UnionAndIntersection [Java Application] C:\Users\Shree\.p2\p
Enter size of two array: 3 4
Enter 1 array element: 2 5 6
Enter 2 array element: 4 6 8 10
Intersection of array element is: 6
Union of array element is: 2 4 5 6 8 10
```



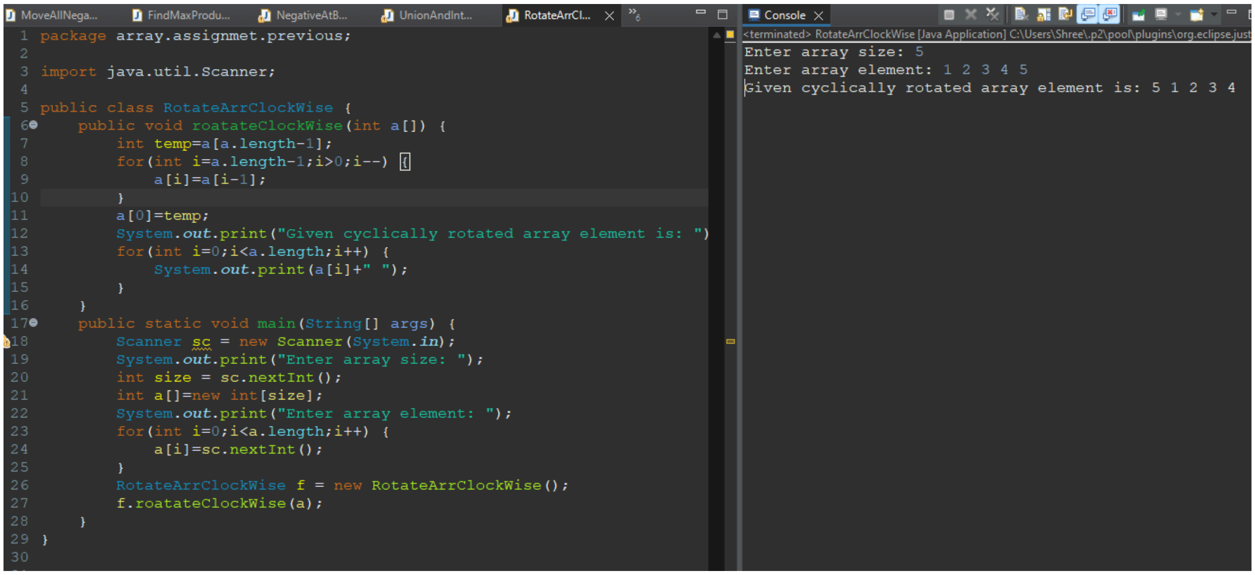
```
        for(int i=0;i<c.length;i++) {
            for(int j=i+1;j<c.length;j++) {
                if(c[i]==c[j]) {
                    if(c[i]!=-1) {
                        System.out.print(c[i]+" ");
                    }
                    if(c[i]==c[j]) {
                        c[j]=-1;
                    }
                }
            }
        }
        System.out.print("\nUnion of array element is: ");
        for(int i=0;i<c.length;i++) {
            if(c[i]!=-1)
            System.out.print(c[i]+" ");
        }
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter size of two array: ");
        int size = sc.nextInt();
        int size1 = sc.nextInt();
        int a[]=new int[size];
        int b[]=new int[size1];
        System.out.print("Enter 1 array element: ");
        for(int i=0;i<a.length;i++) {
            a[i]=sc.nextInt();
        }
        System.out.print("Enter 2 array element: ");
        for(int i=0;i<b.length;i++) {
            b[i]=sc.nextInt();
        }
        UnionAndIntersection f = new UnionAndIntersection();
        f.unionAndInterArrEle(a,b,size,size1);
    }
}
```

```
<terminated> UnionAndIntersection [Java Application] C:\Users\Shree\.p2\pool
Enter size of two array: 3 4
Enter 1 array element: 2 5 6
Enter 2 array element: 4 6 8 10
Intersection of array element is: 6
Union of array element is: 2 4 5 6 8 10
```

Q10. Program to cyclically rotate an array by one

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

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



```java
package array.assignmet.previous;

import java.util.Scanner;

public class RotateArrClockWise {
    public void roatateClockWise(int a[]) {
        int temp=a[a.length-1];
        for(int i=a.length-1;i>0;i--) {
            a[i]=a[i-1];
        }
        a[0]=temp;
        System.out.print("Given cyclically rotated array element is: ")
        for(int i=0;i<a.length;i++) {
            System.out.print(a[i]+" ");
        }
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter array size: ");
        int size = sc.nextInt();
        int a[]=new int[size];
        System.out.print("Enter array element: ");
        for(int i=0;i<a.length;i++) {
            a[i]=sc.nextInt();
        }
        RotateArrClockWise f = new RotateArrClockWise();
        f.roatateClockWise(a);
    }
}
```

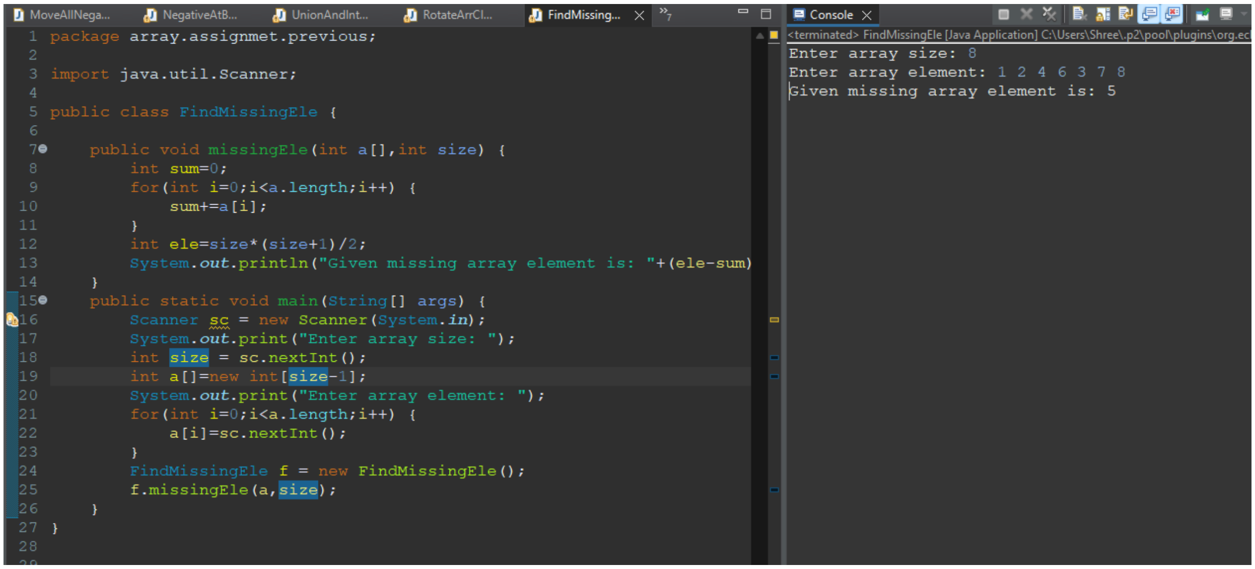
```
<terminated> RotateArrClockWise [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclipse.just
Enter array size: 5
Enter array element: 1 2 3 4 5
Given cyclically rotated array element is: 5 1 2 3 4
```

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



```java
package array.assignmet.previous;

import java.util.Scanner;

public class FindMissingEle {

    public void missingEle(int a[],int size) {
        int sum=0;
        for(int i=0;i<a.length;i++) {
            sum+=a[i];
        }
        int ele=size*(size+1)/2;
        System.out.println("Given missing array element is: "+(ele-sum)
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter array size: ");
        int size = sc.nextInt();
        int a[]=new int[size-1];
        System.out.print("Enter array element: ");
        for(int i=0;i<a.length;i++) {
            a[i]=sc.nextInt();
        }
        FindMissingEle f = new FindMissingEle();
        f.missingEle(a,size);
    }
}
```

```
<terminated> FindMissingEle [Java Application] C:\Users\Shree\.p2\pool\plugins\org.ecl
Enter array size: 8
Enter array element: 1 2 4 6 3 7 8
Given missing array element is: 5
```

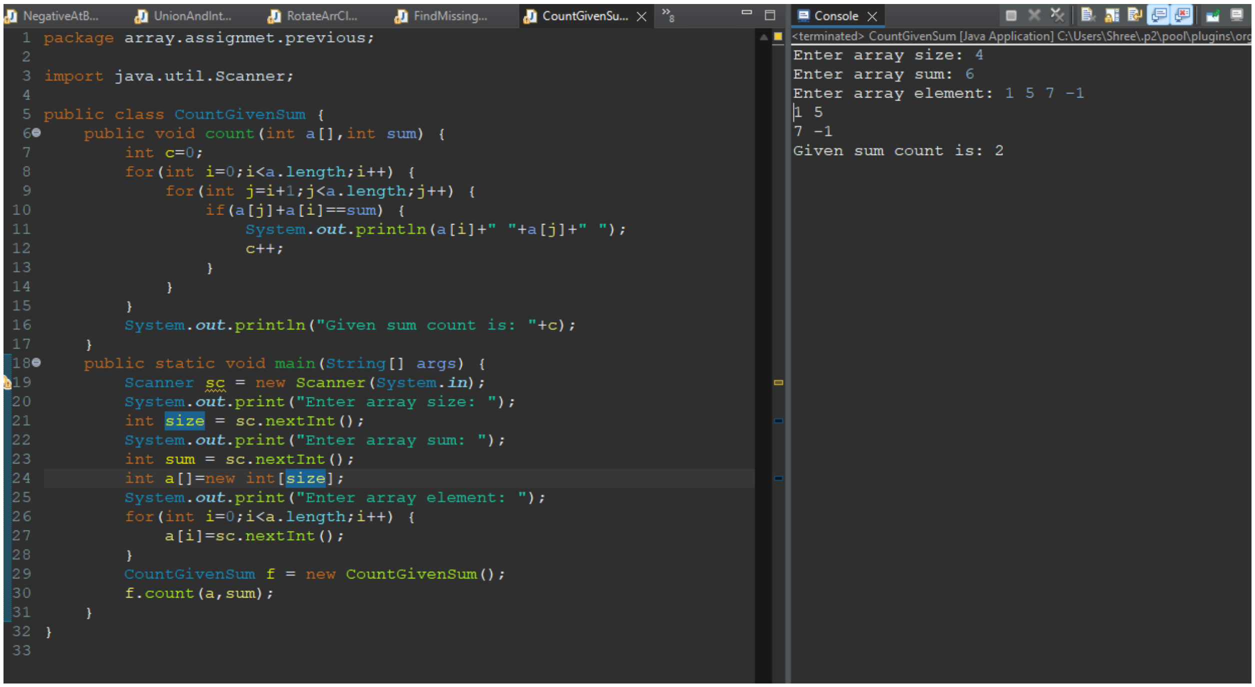
12. Count pairs with given sum

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

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

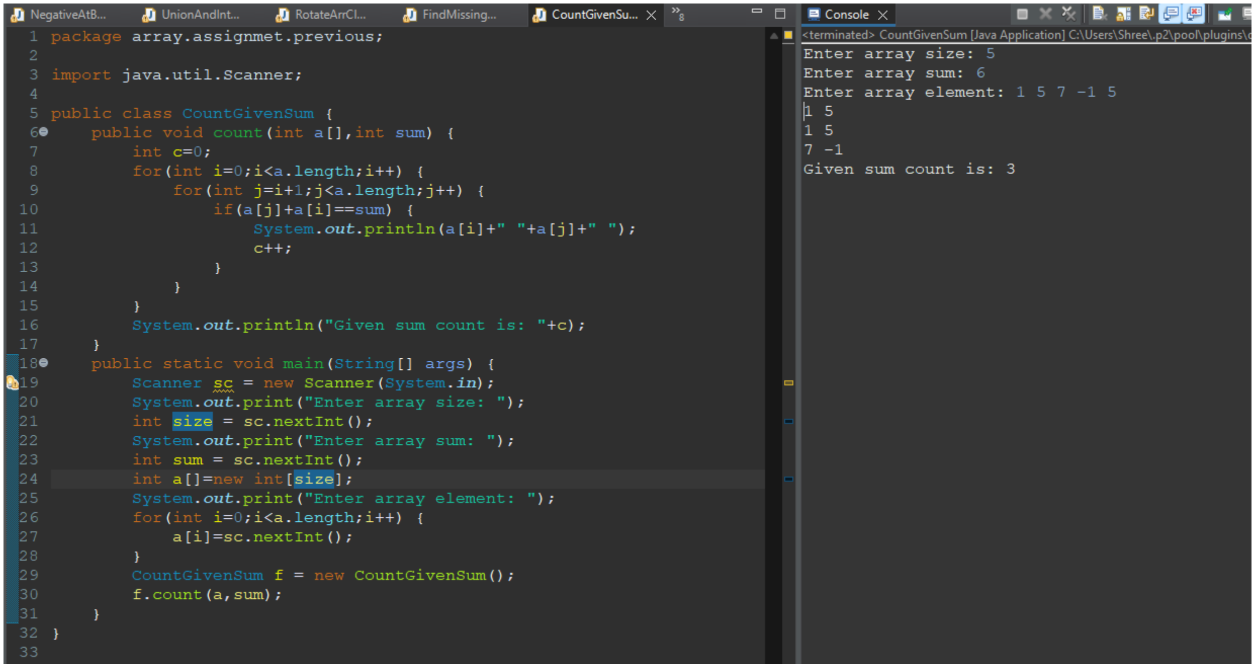
Input: arr[] = {1, 5, 7, -1, 5}, sum = 6

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



```java
package array.assignmet.previous;

import java.util.Scanner;

public class CountGivenSum {
    public void count(int a[],int sum) {
        int c=0;
        for(int i=0;i<a.length;i++) {
            for(int j=i+1;j<a.length;j++) {
                if(a[j]+a[i]==sum) {
                    System.out.println(a[i]+" "+a[j]+" ");
                    c++;
                }
            }
        }
        System.out.println("Given sum count is: "+c);
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter array size: ");
        int size = sc.nextInt();
        System.out.print("Enter array sum: ");
        int sum = sc.nextInt();
        int a[]=new int[size];
        System.out.print("Enter array element: ");
        for(int i=0;i<a.length;i++) {
            a[i]=sc.nextInt();
        }
        CountGivenSum f = new CountGivenSum();
        f.count(a,sum);
    }
}
```

```
<terminated> CountGivenSum [Java Application] C:\Users\Shree\.p2\pool\plugins\or
Enter array size: 4
Enter array sum: 6
Enter array element: 1 5 7 -1
1 5
7 -1
Given sum count is: 2
```



```java
package array.assignmet.previous;

import java.util.Scanner;

public class CountGivenSum {
    public void count(int a[],int sum) {
        int c=0;
        for(int i=0;i<a.length;i++) {
            for(int j=i+1;j<a.length;j++) {
                if(a[j]+a[i]==sum) {
                    System.out.println(a[i]+" "+a[j]+" ");
                    c++;
                }
            }
        }
        System.out.println("Given sum count is: "+c);
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter array size: ");
        int size = sc.nextInt();
        System.out.print("Enter array sum: ");
        int sum = sc.nextInt();
        int a[]=new int[size];
        System.out.print("Enter array element: ");
        for(int i=0;i<a.length;i++) {
            a[i]=sc.nextInt();
        }
        CountGivenSum f = new CountGivenSum();
        f.count(a,sum);
    }
}
```

```
<terminated> CountGivenSum [Java Application] C:\Users\Shree\.p2\pool\plugins\o
Enter array size: 5
Enter array sum: 6
Enter array element: 1 5 7 -1 5
1 5
1 5
7 -1
Given sum count is: 3
```

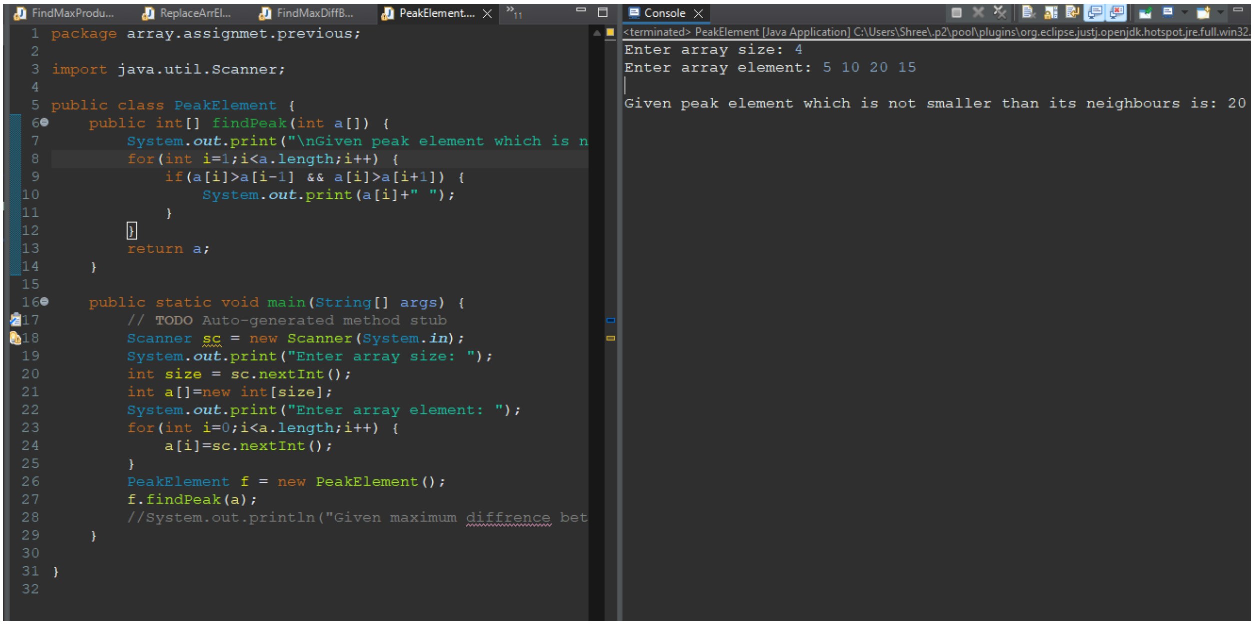
Q1.Given an array arr[] of integers. Find a peak element i.e. an element that is not smaller than its neighbors.

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

Example:

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

Output: 20

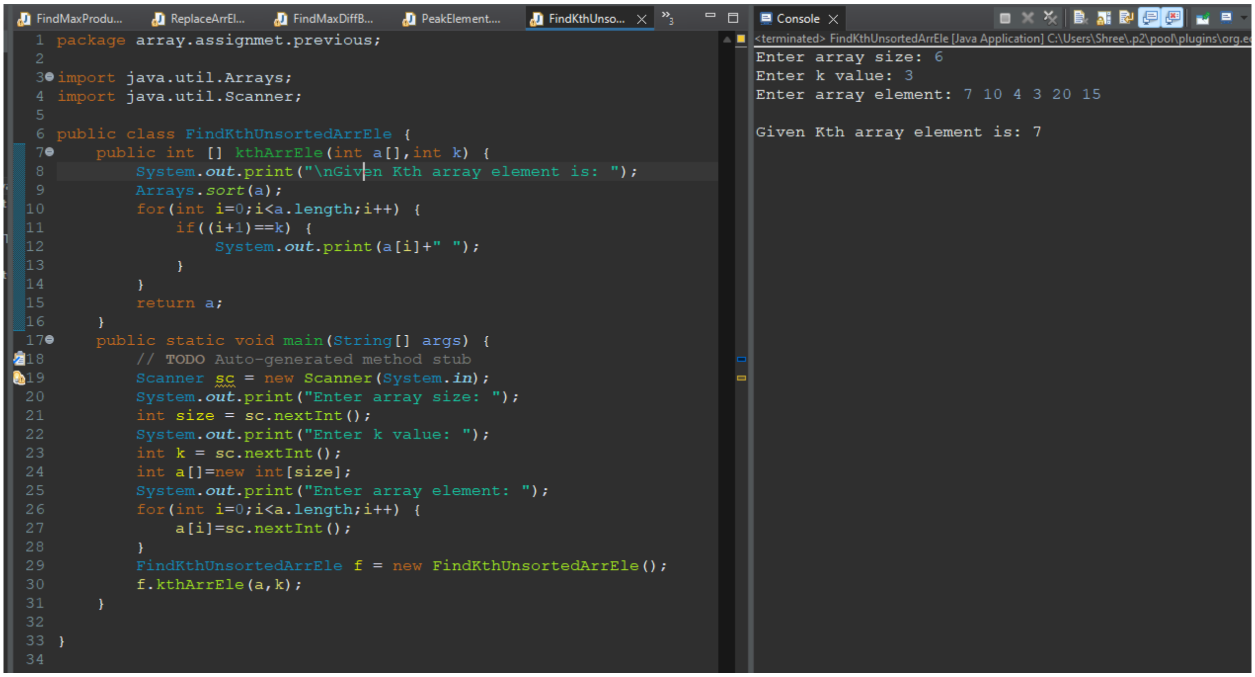


```java
package array.assignmet.previous;

import java.util.Scanner;

public class PeakElement {
    public int[] findPeak(int a[]) {
        System.out.print("\nGiven peak element which is n
        for(int i=1;i<a.length;i++) {
            if(a[i]>a[i-1] && a[i]>a[i+1]) {
                System.out.print(a[i]+" ");
            }
        }
        return a;
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter array size: ");
        int size = sc.nextInt();
        int a[]=new int[size];
        System.out.print("Enter array element: ");
        for(int i=0;i<a.length;i++) {
            a[i]=sc.nextInt();
        }
        PeakElement f = new PeakElement();
        f.findPeak(a);
        //System.out.println("Given maximum diffrence bet
    }

}
```

```
<terminated> PeakElement [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.
Enter array size: 4
Enter array element: 5 10 20 15

Given peak element which is not smaller than its neighbours is: 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


FindMaxProdu...
ReplaceArrEl...
FindMaxDiffB...
PeakElement....
FindKthUnso...
Console
<terminated> FindKthUnsortedArrEle [Java Application] C:\Users\Shree\.p2\pool\plugins\org.e

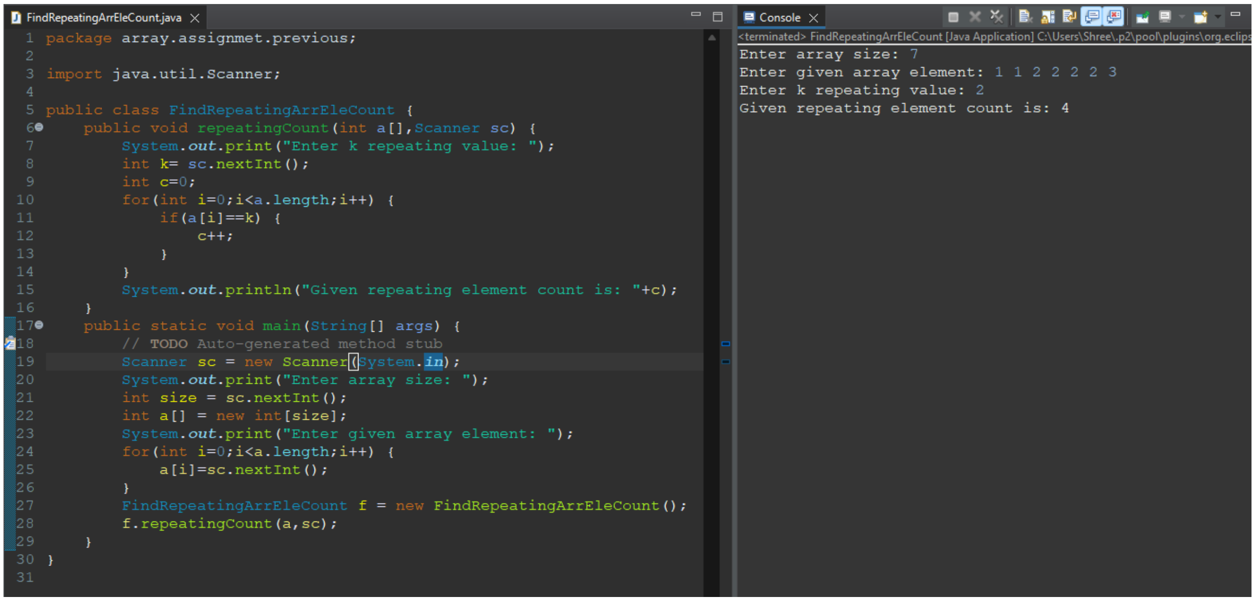
```java
package array.assignmet.previous;

import java.util.Arrays;
import java.util.Scanner;

public class FindKthUnsortedArrEle {
    public int [] kthArrEle(int a[],int k) {
        System.out.print("\nGiven Kth array element is: ");
        Arrays.sort(a);
        for(int i=0;i<a.length;i++) {
            if((i+1)==k) {
                System.out.print(a[i]+" ");
            }
        }
        return a;
    }
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter array size: ");
        int size = sc.nextInt();
        System.out.print("Enter k value: ");
        int k = sc.nextInt();
        int a[]=new int[size];
        System.out.print("Enter array element: ");
        for(int i=0;i<a.length;i++) {
            a[i]=sc.nextInt();
        }
        FindKthUnsortedArrEle f = new FindKthUnsortedArrEle();
        f.kthArrEle(a,k);
    }

}
```

```
Enter array size: 6
Enter k value: 3
Enter array element: 7 10 4 3 20 15

Given Kth array element is: 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

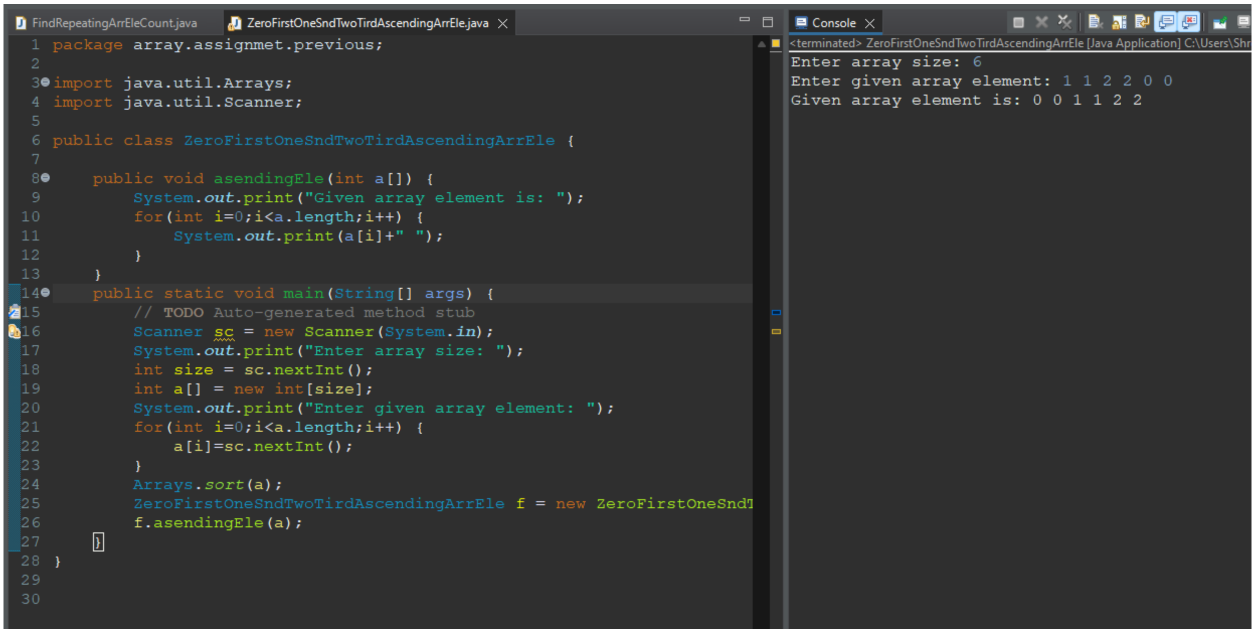
Output: 4 // x (or 2) occurs 4 times in arr[]


FindRepeatingArrEleCount.java
Console
<terminated> FindRepeatingArrEleCount [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclips

```java
package array.assignmet.previous;

import java.util.Scanner;

public class FindRepeatingArrEleCount {
    public void repeatingCount(int a[],Scanner sc) {
        System.out.print("Enter k repeating value: ");
        int k= sc.nextInt();
        int c=0;
        for(int i=0;i<a.length;i++) {
            if(a[i]==k) {
                c++;
            }
        }
        System.out.println("Given repeating element count is: "+c);
    }
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter array size: ");
        int size = sc.nextInt();
        int a[] = new int[size];
        System.out.print("Enter given array element: ");
        for(int i=0;i<a.length;i++) {
            a[i]=sc.nextInt();
        }
        FindRepeatingArrEleCount f = new FindRepeatingArrEleCount();
        f.repeatingCount(a,sc);
    }
}
```

```
Enter array size: 7
Enter given array element: 1 1 2 2 2 2 3
Enter k repeating value: 2
Given repeating element count is: 4
```

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.


FindRepeatingArrEleCount.java
ZeroFirstOneSndTwoTirdAscendingArrEle.java
Console
<terminated> ZeroFirstOneSndTwoTirdAscendingArrEle [Java Application] C:\Users\Shr

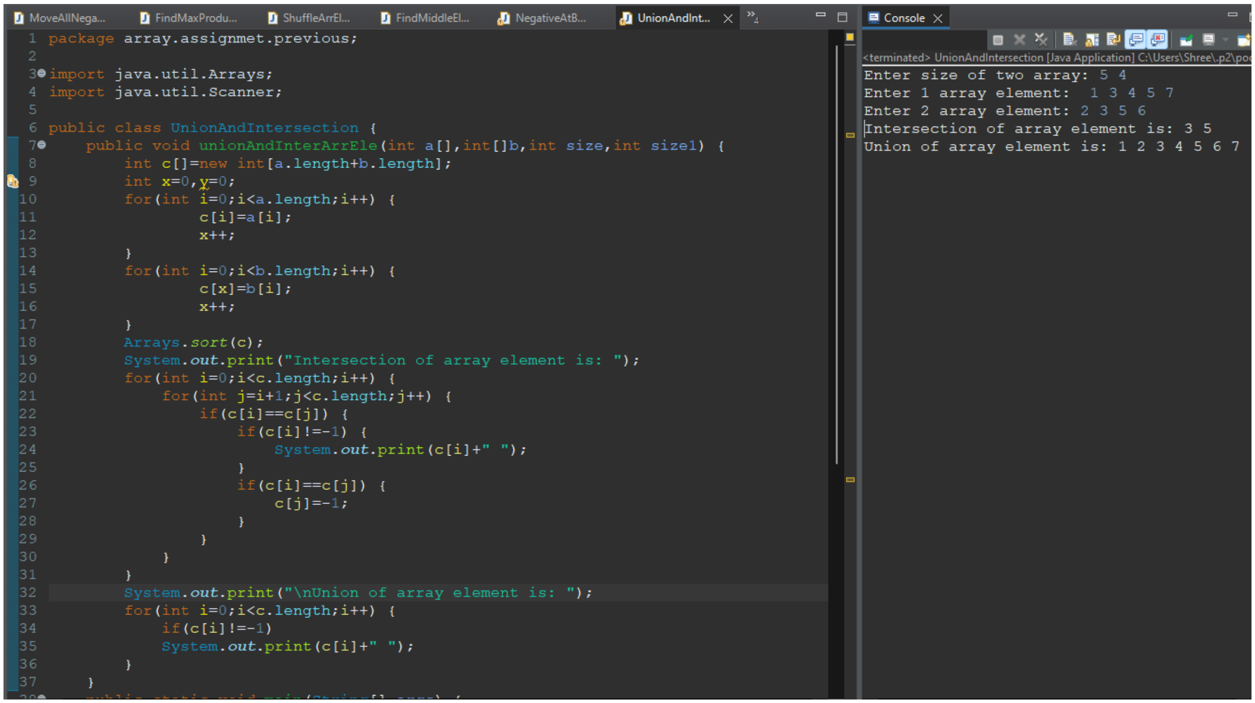
```java
package array.assignmet.previous;

import java.util.Arrays;
import java.util.Scanner;

public class ZeroFirstOneSndTwoTirdAscendingArrEle {

    public void asendingEle(int a[]) {
        System.out.print("Given array element is: ");
        for(int i=0;i<a.length;i++) {
            System.out.print(a[i]+" ");
        }
    }
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter array size: ");
        int size = sc.nextInt();
        int a[] = new int[size];
        System.out.print("Enter given array element: ");
        for(int i=0;i<a.length;i++) {
            a[i]=sc.nextInt();
        }
        Arrays.sort(a);
        ZeroFirstOneSndTwoTirdAscendingArrEle f = new ZeroFirstOneSndT
        f.asendingEle(a);
    }
}
```

```
Enter array size: 6
Enter given array element: 1 1 2 2 0 0
Given array element is: 0 0 1 1 2 2
```

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

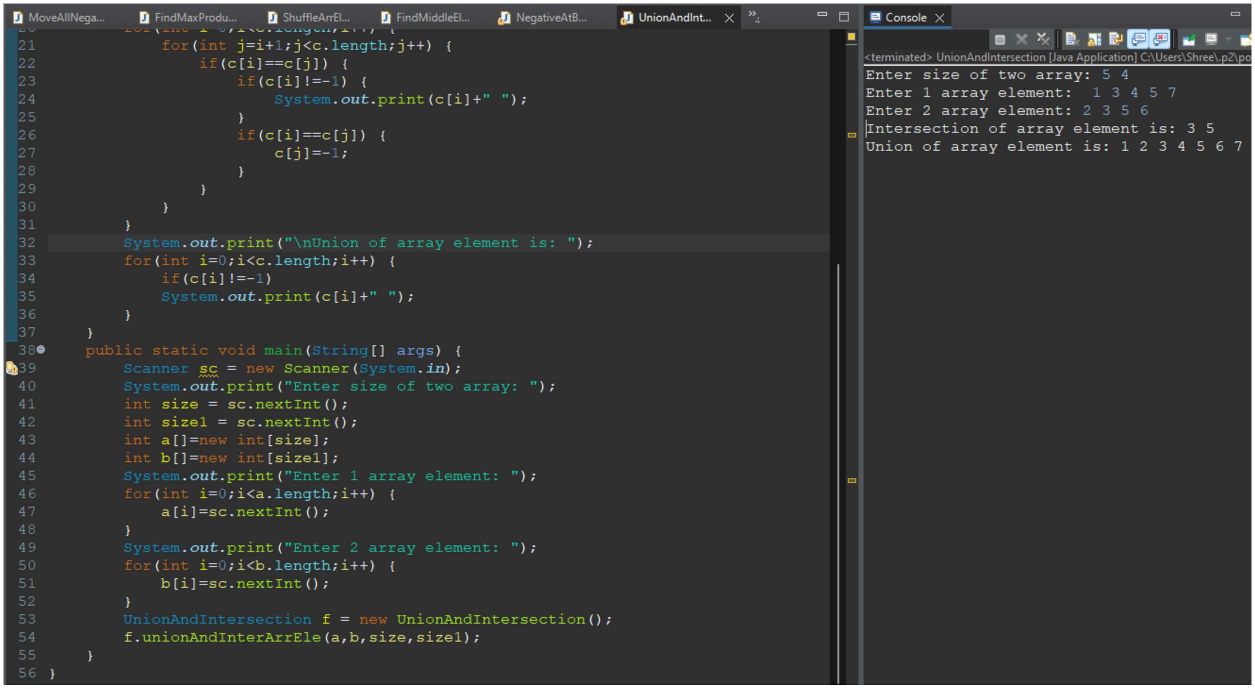
Example: Input: arr1[] = {1, 3, 4, 5, 7}   arr2[] = {2, 3, 5, 6}

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



```
package array.assignmet.previous;

import java.util.Arrays;
import java.util.Scanner;

public class UnionAndIntersection {
    public void unionAndInterArrEle(int a[],int[]b,int size,int size1) {
        int c[]=new int[a.length+b.length];
        int x=0,y=0;
        for(int i=0;i<a.length;i++) {
                c[i]=a[i];
                x++;
        }
        for(int i=0;i<b.length;i++) {
                c[x]=b[i];
                x++;
        }
        Arrays.sort(c);
        System.out.print("Intersection of array element is: ");
        for(int i=0;i<c.length;i++) {
            for(int j=i+1;j<c.length;j++) {
                if(c[i]==c[j]) {
                    if(c[i]!=-1) {
                        System.out.print(c[i]+" ");
                    }
                    if(c[i]==c[j]) {
                        c[j]=-1;
                    }
                }
            }
        }
        System.out.print("\nUnion of array element is: ");
        for(int i=0;i<c.length;i++) {
            if(c[i]!=-1)
            System.out.print(c[i]+" ");
        }
    }
```

```
<terminated> UnionAndIntersection [Java Application] C:\Users\Shree\.p2\poo
Enter size of two array: 5 4
Enter 1 array element:  1 3 4 5 7
Enter 2 array element: 2 3 5 6
Intersection of array element is: 3 5
Union of array element is: 1 2 3 4 5 6 7
```



```
            for(int j=i+1;j<c.length;j++) {
                if(c[i]==c[j]) {
                    if(c[i]!=-1) {
                        System.out.print(c[i]+" ");
                    }
                    if(c[i]==c[j]) {
                        c[j]=-1;
                    }
                }
            }
        }
        System.out.print("\nUnion of array element is: ");
        for(int i=0;i<c.length;i++) {
            if(c[i]!=-1)
            System.out.print(c[i]+" ");
        }
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter size of two array: ");
        int size = sc.nextInt();
        int size1 = sc.nextInt();
        int a[]=new int[size];
        int b[]=new int[size1];
        System.out.print("Enter 1 array element: ");
        for(int i=0;i<a.length;i++) {
            a[i]=sc.nextInt();
        }
        System.out.print("Enter 2 array element: ");
        for(int i=0;i<b.length;i++) {
            b[i]=sc.nextInt();
        }
        UnionAndIntersection f = new UnionAndIntersection();
        f.unionAndInterArrEle(a,b,size,size1);
    }
}
```

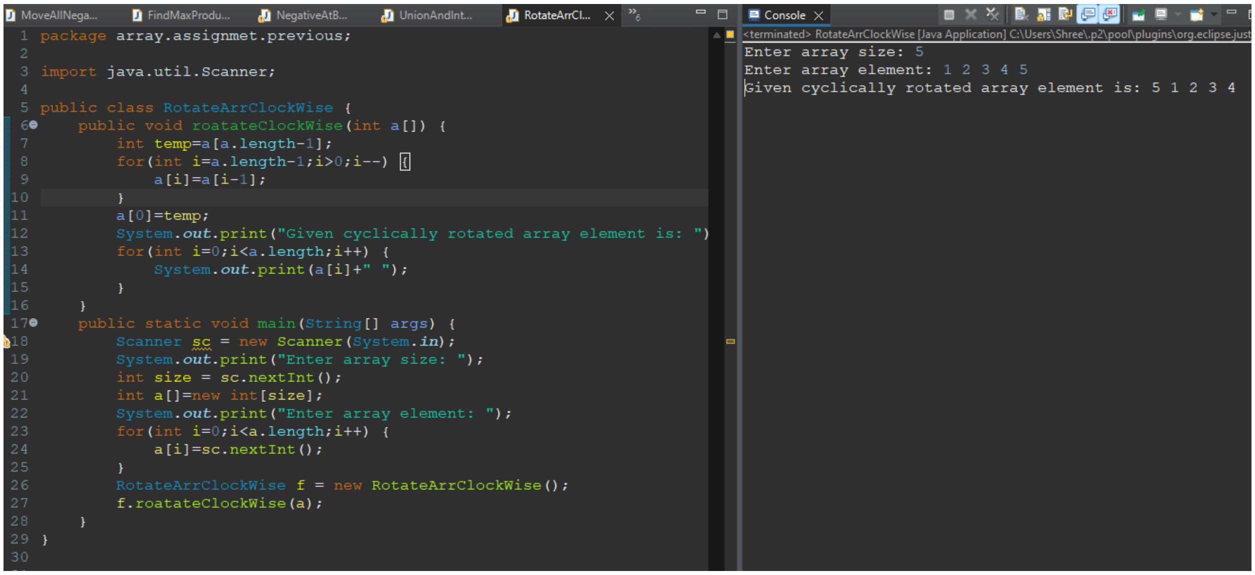
```
<terminated> UnionAndIntersection [Java Application] C:\Users\Shree\.p2\poo
Enter size of two array: 5 4
Enter 1 array element:  1 3 4 5 7
Enter 2 array element: 2 3 5 6
Intersection of array element is: 3 5
Union of array element is: 1 2 3 4 5 6 7
```

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}

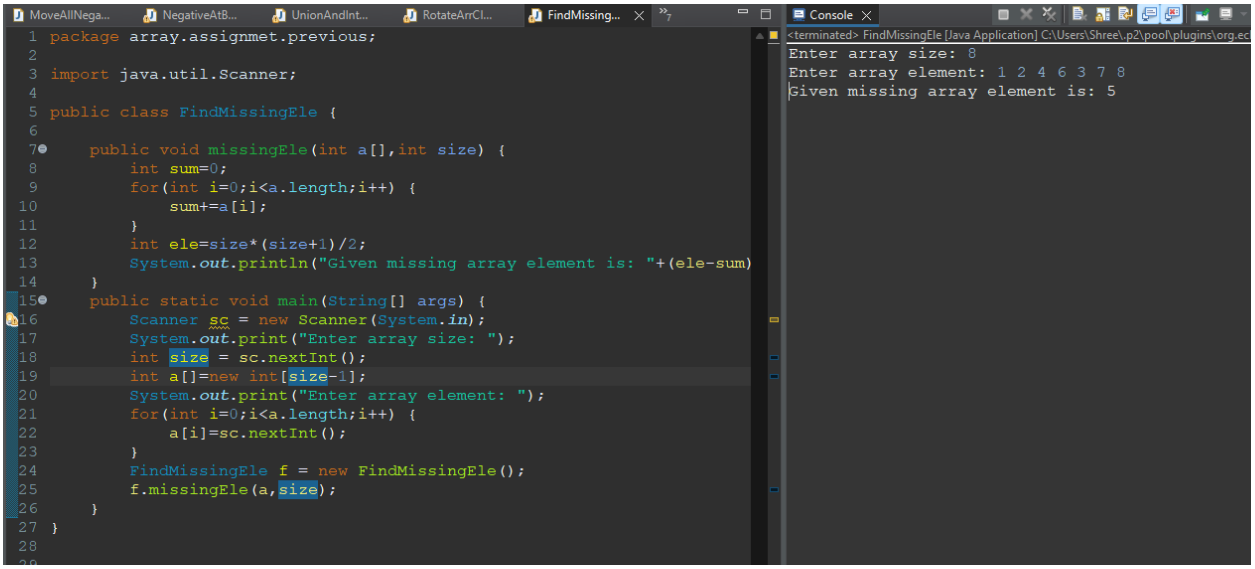


```java
package array.assignmet.previous;

import java.util.Scanner;

public class RotateArrClockWise {
    public void roatateClockWise(int a[]) {
        int temp=a[a.length-1];
        for(int i=a.length-1;i>0;i--) {
            a[i]=a[i-1];
        }
        a[0]=temp;
        System.out.print("Given cyclically rotated array element is: ")
        for(int i=0;i<a.length;i++) {
            System.out.print(a[i]+" ");
        }
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter array size: ");
        int size = sc.nextInt();
        int a[]=new int[size];
        System.out.print("Enter array element: ");
        for(int i=0;i<a.length;i++) {
            a[i]=sc.nextInt();
        }
        RotateArrClockWise f = new RotateArrClockWise();
        f.roatateClockWise(a);
    }
}
```

```
<terminated> RotateArrClockWise [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclipse.just
Enter array size: 5
Enter array element: 1 2 3 4 5
Given cyclically rotated array element is: 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



```java
package array.assignmet.previous;

import java.util.Scanner;

public class FindMissingEle {

    public void missingEle(int a[],int size) {
        int sum=0;
        for(int i=0;i<a.length;i++) {
            sum+=a[i];
        }
        int ele=size*(size+1)/2;
        System.out.println("Given missing array element is: "+(ele-sum)
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter array size: ");
        int size = sc.nextInt();
        int a[]=new int[size-1];
        System.out.print("Enter array element: ");
        for(int i=0;i<a.length;i++) {
            a[i]=sc.nextInt();
        }
        FindMissingEle f = new FindMissingEle();
        f.missingEle(a,size);
    }
}
```

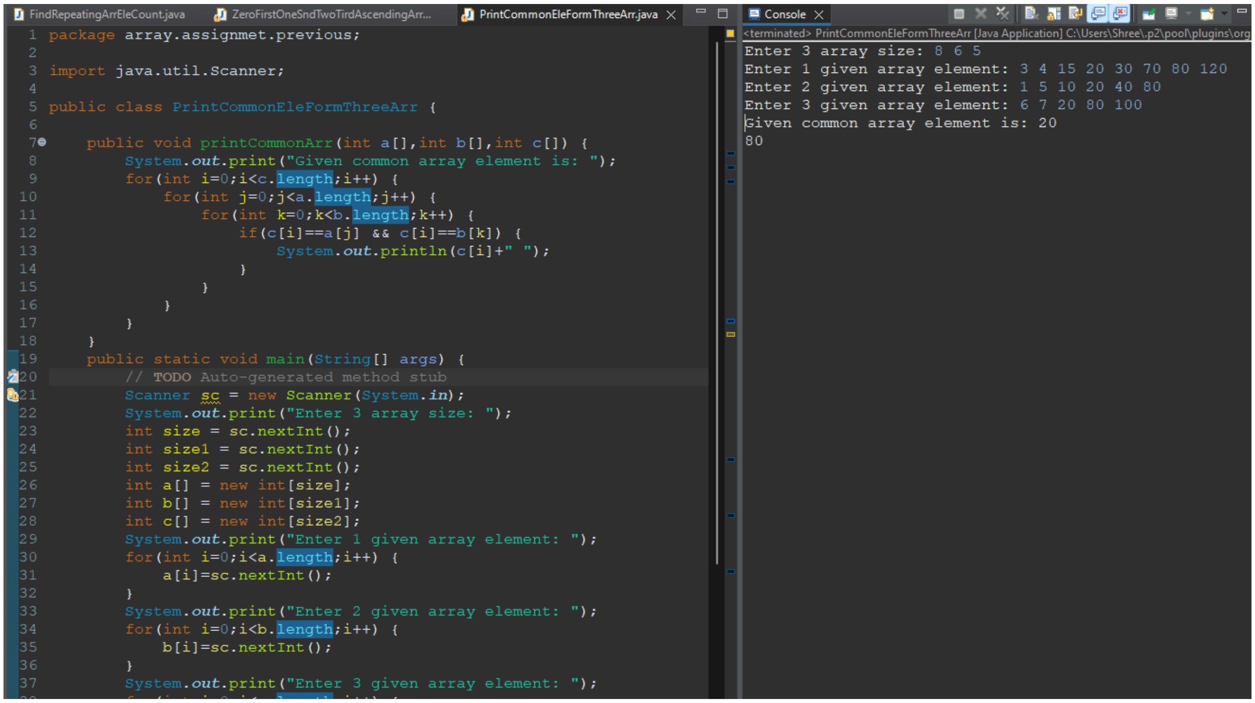
```
<terminated> FindMissingEle [Java Application] C:\Users\Shree\.p2\pool\plugins\org.ecl
Enter array size: 8
Enter array element: 1 2 4 6 3 7 8
Given missing array element is: 5
```

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

Examples:

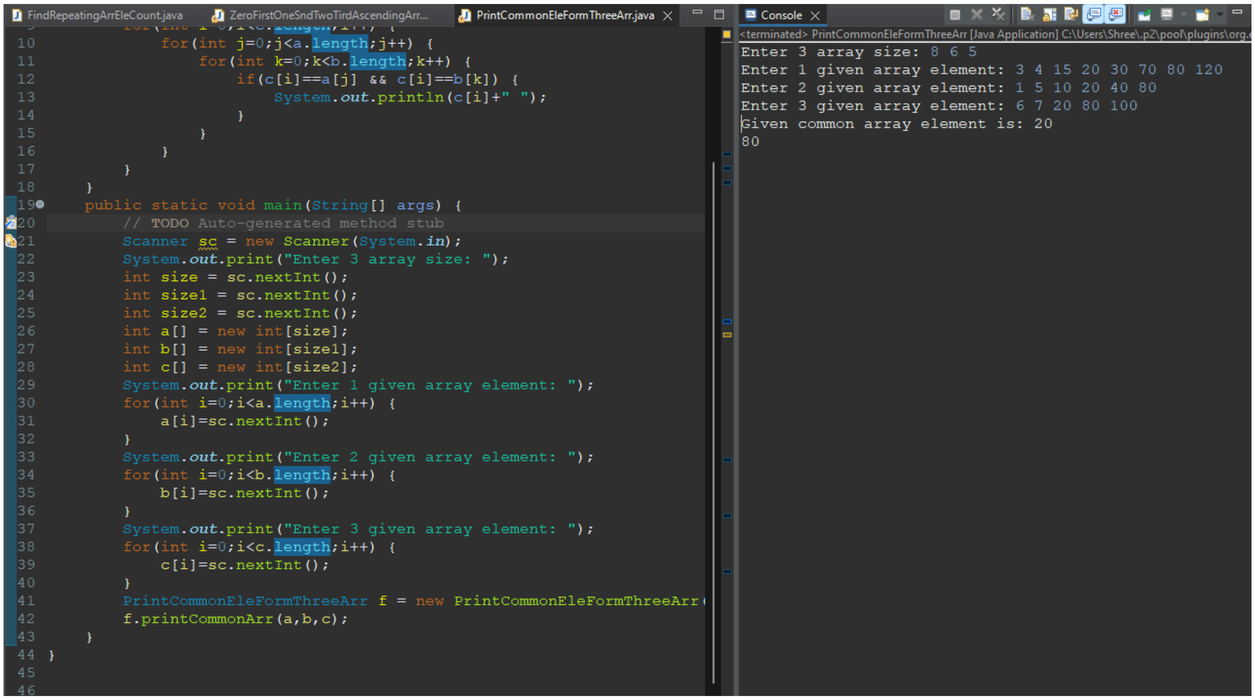
Input: ar1[] = {1, 5, 10, 20, 40, 80} ar2[] = {6, 7, 20, 80, 100} ar3[] = {3, 4, 15, 20, 30, 70, 80, 120}

Output: 20, 80


FindRepeatingArrEleCount.java
ZeroFirstOneSndTwoTirdAscendingArr...
PrintCommonEleFormThreeArr.java
Console
<terminated> PrintCommonEleFormThreeArr [Java Application] C:\Users\Shree\.p2\pool\plugins\org

```java
package array.assignmet.previous;

import java.util.Scanner;

public class PrintCommonEleFormThreeArr {

    public void printCommonArr(int a[],int b[],int c[]) {
        System.out.print("Given common array element is: ");
        for(int i=0;i<c.length;i++) {
            for(int j=0;j<a.length;j++) {
                for(int k=0;k<b.length;k++) {
                    if(c[i]==a[j] && c[i]==b[k]) {
                        System.out.println(c[i]+" ");
                    }
                }
            }
        }
    }
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter 3 array size: ");
        int size = sc.nextInt();
        int size1 = sc.nextInt();
        int size2 = sc.nextInt();
        int a[] = new int[size];
        int b[] = new int[size1];
        int c[] = new int[size2];
        System.out.print("Enter 1 given array element: ");
        for(int i=0;i<a.length;i++) {
            a[i]=sc.nextInt();
        }
        System.out.print("Enter 2 given array element: ");
        for(int i=0;i<b.length;i++) {
            b[i]=sc.nextInt();
        }
        System.out.print("Enter 3 given array element: ");
```

```
Enter 3 array size: 8 6 5
Enter 1 given array element: 3 4 15 20 30 70 80 120
Enter 2 given array element: 1 5 10 20 40 80
Enter 3 given array element: 6 7 20 80 100
Given common array element is: 20
80
```


FindRepeatingArrEleCount.java
ZeroFirstOneSndTwoTirdAscendingArr...
PrintCommonEleFormThreeArr.java
Console
<terminated> PrintCommonEleFormThreeArr [Java Application] C:\Users\Shree\.p2\pool\plugins\org.

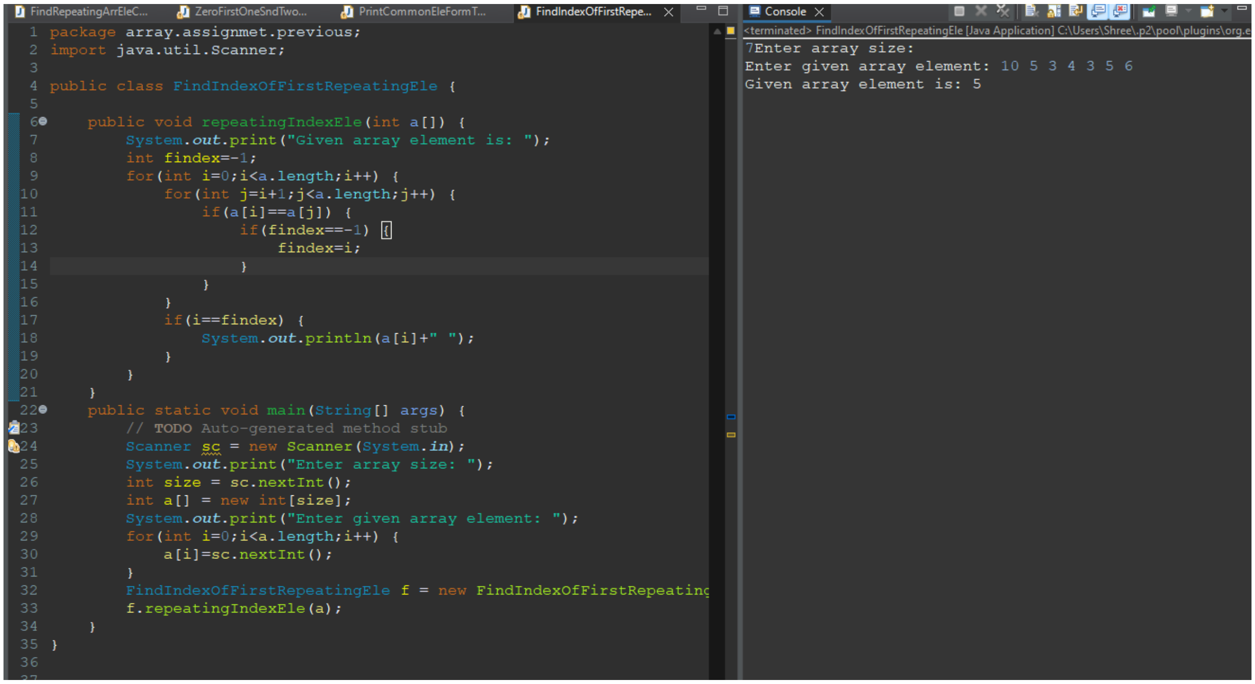
```java
            for(int j=0;j<a.length;j++) {
                for(int k=0;k<b.length;k++) {
                    if(c[i]==a[j] && c[i]==b[k]) {
                        System.out.println(c[i]+" ");
                    }
                }
            }
        }
    }
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter 3 array size: ");
        int size = sc.nextInt();
        int size1 = sc.nextInt();
        int size2 = sc.nextInt();
        int a[] = new int[size];
        int b[] = new int[size1];
        int c[] = new int[size2];
        System.out.print("Enter 1 given array element: ");
        for(int i=0;i<a.length;i++) {
            a[i]=sc.nextInt();
        }
        System.out.print("Enter 2 given array element: ");
        for(int i=0;i<b.length;i++) {
            b[i]=sc.nextInt();
        }
        System.out.print("Enter 3 given array element: ");
```

```java
        for(int i=0;i<c.length;i++) {
            c[i]=sc.nextInt();
        }
        PrintCommonEleFormThreeArr f = new PrintCommonEleFormThreeArr(
        f.printCommonArr(a,b,c);
    }
}
```

```
Enter 3 array size: 8 6 5
Enter 1 given array element: 3 4 15 20 30 70 80 120
Enter 2 given array element: 1 5 10 20 40 80
Enter 3 given array element: 6 7 20 80 100
Given common array element is: 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



```java
package array.assignmet.previous;
import java.util.Scanner;

public class FindIndexOfFirstRepeatingEle {

    public void repeatingIndexEle(int a[]) {
        System.out.print("Given array element is: ");
        int findex=-1;
        for(int i=0;i<a.length;i++) {
            for(int j=i+1;j<a.length;j++) {
                if(a[i]==a[j]) {
                    if(findex==-1) {
                        findex=i;
                    }
                }
            }
            if(i==findex) {
                System.out.println(a[i]+" ");
            }
        }
    }
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter array size: ");
        int size = sc.nextInt();
        int a[] = new int[size];
        System.out.print("Enter given array element: ");
        for(int i=0;i<a.length;i++) {
            a[i]=sc.nextInt();
        }
        FindIndexOfFirstRepeatingEle f = new FindIndexOfFirstRepeating
        f.repeatingIndexEle(a);
    }
}
```

```
<terminated> FindIndexOfFirstRepeatingEle [Java Application] C:\Users\Shree\.p2\pool\plugins\org.e
7Enter array size:
Enter given array element: 10 5 3 4 3 5 6
Given array element is: 5
```

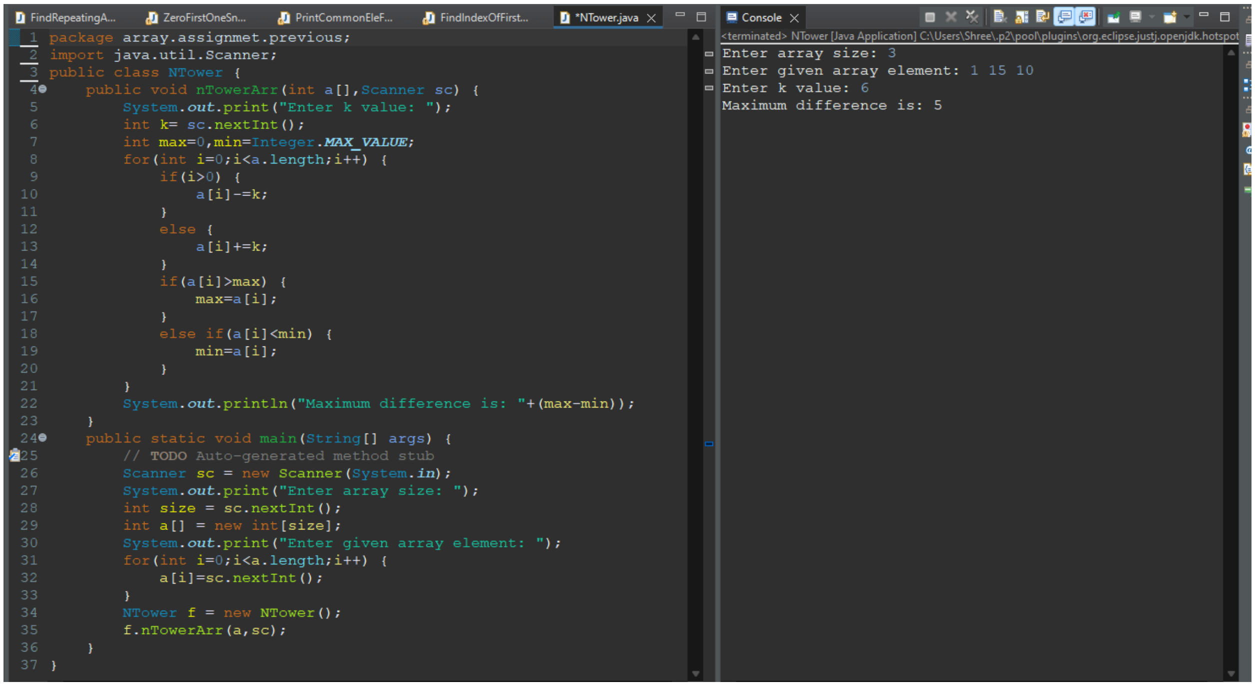
Q10.Given the heights of N towers and a value of K, Either increase or decrease the height of every tower by K (only once) where K > 0. After modifications,

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.



```java
package array.assignmet.previous;
import java.util.Scanner;
public class NTower {
    public void nTowerArr(int a[],Scanner sc) {
        System.out.print("Enter k value: ");
        int k= sc.nextInt();
        int max=0,min=Integer.MAX_VALUE;
        for(int i=0;i<a.length;i++) {
            if(i>0) {
                a[i]-=k;
            }
            else {
                a[i]+=k;
            }
            if(a[i]>max) {
                max=a[i];
            }
            else if(a[i]<min) {
                min=a[i];
            }
        }
        System.out.println("Maximum difference is: "+(max-min));
    }
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter array size: ");
        int size = sc.nextInt();
        int a[] = new int[size];
        System.out.print("Enter given array element: ");
        for(int i=0;i<a.length;i++) {
            a[i]=sc.nextInt();
        }
        NTower f = new NTower();
        f.nTowerArr(a,sc);
    }
}
```

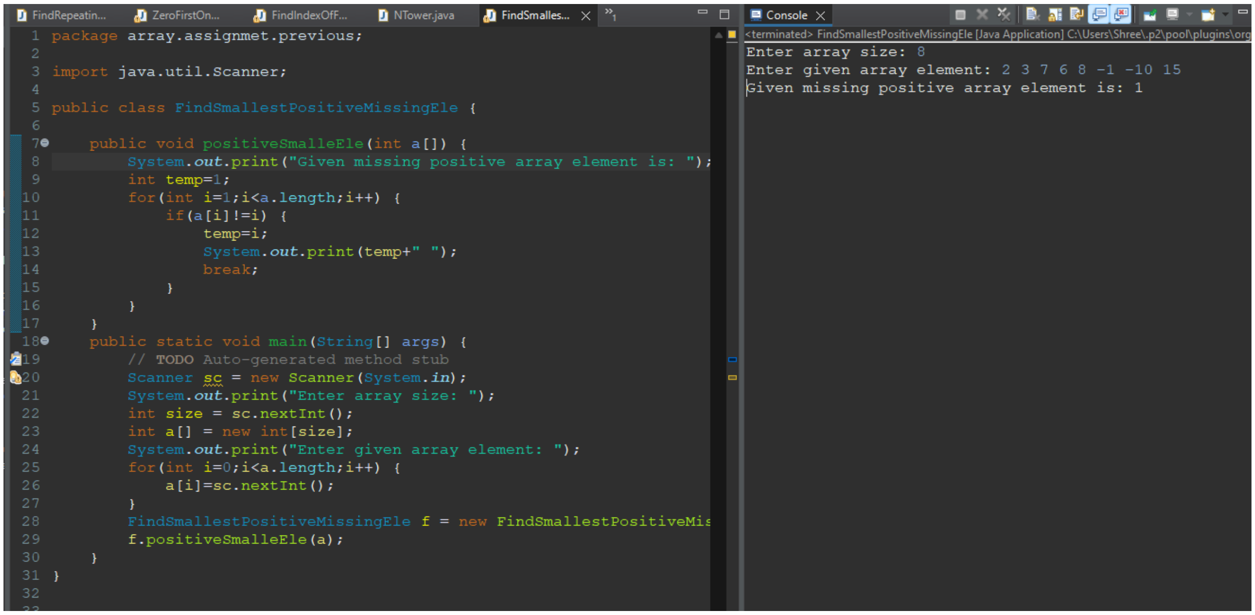
```
<terminated> NTower [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot
Enter array size: 3
Enter given array element: 1 15 10
Enter k value: 6
Maximum difference is: 5
```

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



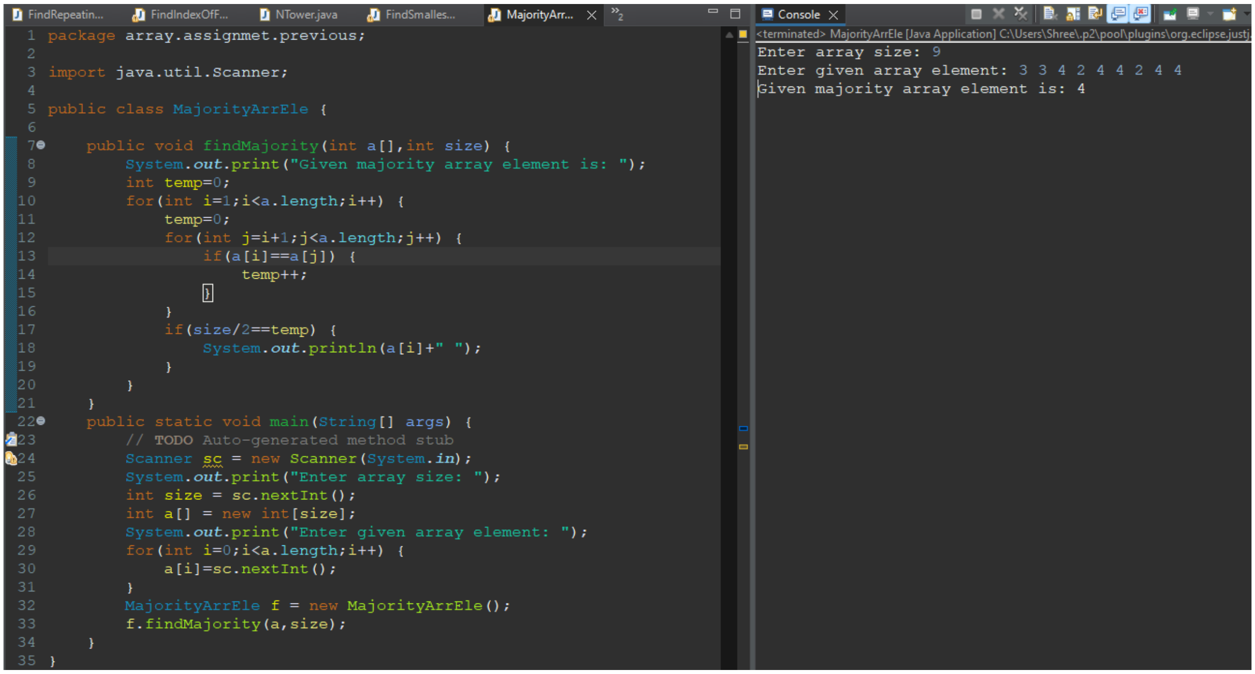
```java
package array.assignmet.previous;

import java.util.Scanner;

public class FindSmallestPositiveMissingEle {

    public void positiveSmalleEle(int a[]) {
        System.out.print("Given missing positive array element is: ");
        int temp=1;
        for(int i=1;i<a.length;i++) {
            if(a[i]!=i) {
                temp=i;
                System.out.print(temp+" ");
                break;
            }
        }
    }
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter array size: ");
        int size = sc.nextInt();
        int a[] = new int[size];
        System.out.print("Enter given array element: ");
        for(int i=0;i<a.length;i++) {
            a[i]=sc.nextInt();
        }
        FindSmallestPositiveMissingEle f = new FindSmallestPositiveMis
        f.positiveSmalleEle(a);
    }
}
```

```
<terminated> FindSmallestPositiveMissingEle [Java Application] C:\Users\Shree\.p2\pool\plugins\org
Enter array size: 8
Enter given array element: 2 3 7 6 8 -1 -10 15
Given missing positive array element is: 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



```java
package array.assignmet.previous;

import java.util.Scanner;

public class MajorityArrEle {

    public void findMajority(int a[],int size) {
        System.out.print("Given majority array element is: ");
        int temp=0;
        for(int i=1;i<a.length;i++) {
            temp=0;
            for(int j=i+1;j<a.length;j++) {
                if(a[i]==a[j]) {
                    temp++;
                }
            }
            if(size/2==temp) {
                System.out.println(a[i]+" ");
            }
        }
    }
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter array size: ");
        int size = sc.nextInt();
        int a[] = new int[size];
        System.out.print("Enter given array element: ");
        for(int i=0;i<a.length;i++) {
            a[i]=sc.nextInt();
        }
        MajorityArrEle f = new MajorityArrEle();
        f.findMajority(a,size);
    }
}
```

```
<terminated> MajorityArrEle [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclipse.justj.
Enter array size: 9
Enter given array element: 3 3 4 2 4 4 2 4 4
Given majority array element is: 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:

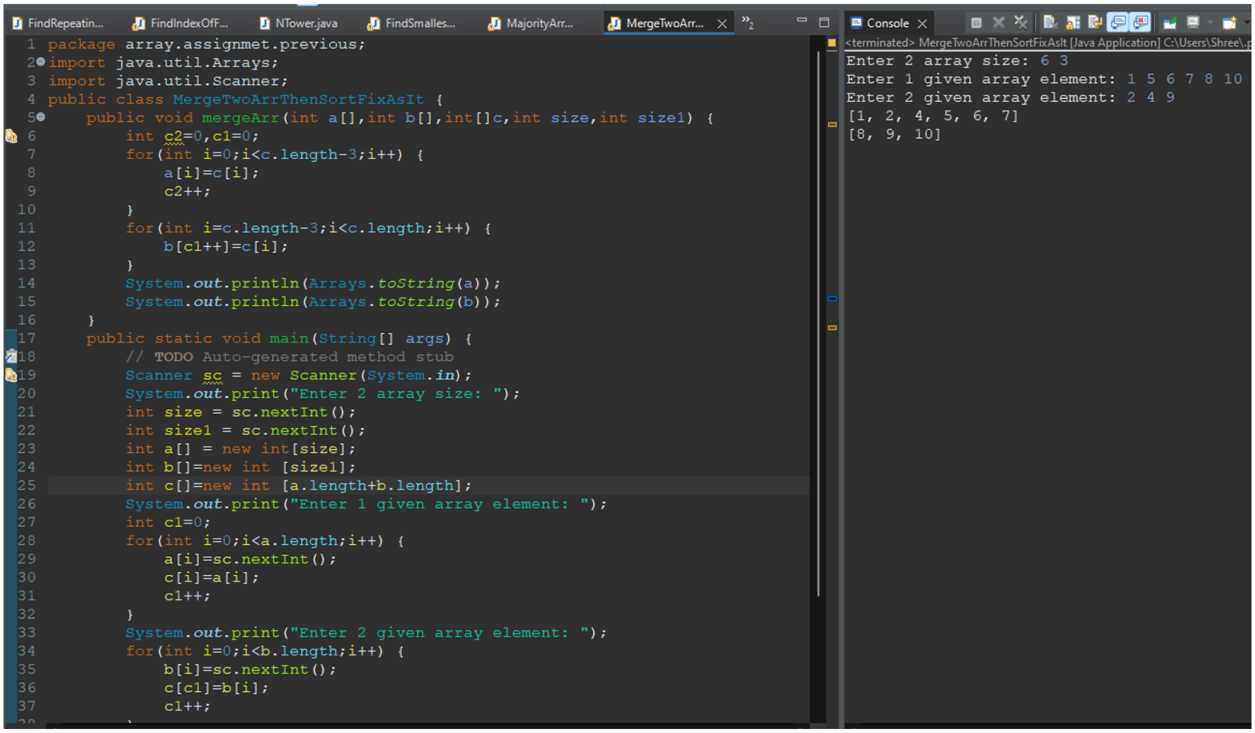
Input: int[] A = { 1, 5, 6, 7, 8, 10 } int[] B = { 2, 4, 9 }

Output:

Sorted Arrays:

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

B: [8, 9, 10]



```java
package array.assignmet.previous;
import java.util.Arrays;
import java.util.Scanner;
public class MergeTwoArrThenSortFixAsIt {
    public void mergeArr(int a[],int b[],int[]c,int size,int size1) {
        int c2=0,c1=0;
        for(int i=0;i<c.length-3;i++) {
            a[i]=c[i];
            c2++;
        }
        for(int i=c.length-3;i<c.length;i++) {
            b[c1++]=c[i];
        }
        System.out.println(Arrays.toString(a));
        System.out.println(Arrays.toString(b));
    }
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter 2 array size: ");
        int size = sc.nextInt();
        int size1 = sc.nextInt();
        int a[] = new int[size];
        int b[]=new int [size1];
        int c[]=new int [a.length+b.length];
        System.out.print("Enter 1 given array element: ");
        int c1=0;
        for(int i=0;i<a.length;i++) {
            a[i]=sc.nextInt();
            c[i]=a[i];
            c1++;
        }
        System.out.print("Enter 2 given array element: ");
        for(int i=0;i<b.length;i++) {
            b[i]=sc.nextInt();
            c[c1]=b[i];
            c1++;
        }
```

```
<terminated> MergeTwoArrThenSortFixAsIt [Java Application] C:\Users\Shree\.p
Enter 2 array size: 6 3
Enter 1 given array element: 1 5 6 7 8 10
Enter 2 given array element: 2 4 9
[1, 2, 4, 5, 6, 7]
[8, 9, 10]
```

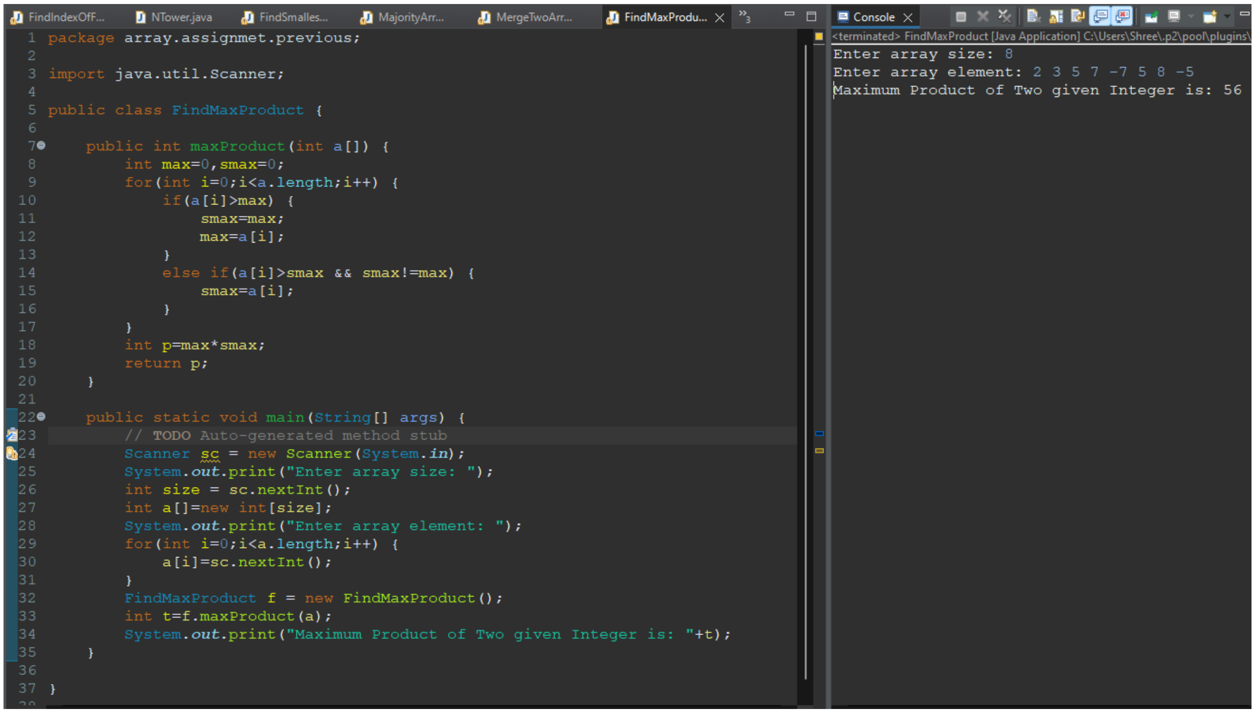
Q14.Write a Java program to find maximum product of two integers in a given array of integers.

Example:

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

Output:

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



```java
package array.assignmet.previous;

import java.util.Scanner;

public class FindMaxProduct {

    public int maxProduct(int a[]) {
        int max=0,smax=0;
        for(int i=0;i<a.length;i++) {
            if(a[i]>max) {
                smax=max;
                max=a[i];
            }
            else if(a[i]>smax && smax!=max) {
                smax=a[i];
            }
        }
        int p=max*smax;
        return p;
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter array size: ");
        int size = sc.nextInt();
        int a[]=new int[size];
        System.out.print("Enter array element: ");
        for(int i=0;i<a.length;i++) {
            a[i]=sc.nextInt();
        }
        FindMaxProduct f = new FindMaxProduct();
        int t=f.maxProduct(a);
        System.out.print("Maximum Product of Two given Integer is: "+t);
    }

}
```

```
<terminated> FindMaxProduct [Java Application] C:\Users\Shree\.p2\pool\plugins\
Enter array size: 8
Enter array element: 2 3 5 7 -7 5 8 -5
Maximum Product of Two given Integer is: 56
```