

# Assignment No:-17

Name:-Suryawanshi Sangramsingh Sambhaji

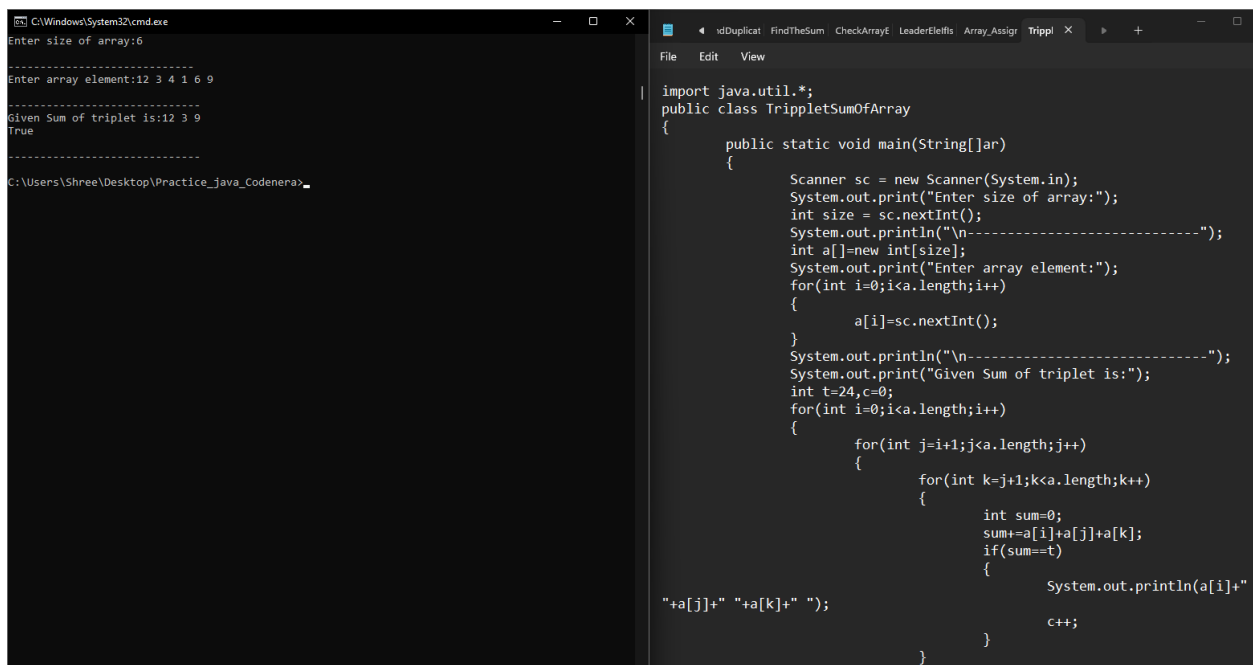
Batch: - Delta - DCA (Java) 2024      Date:-27/5/2024

**1. Given an array and a value, find if there is a triplet in array whose sum is equal to the given value.**

**If there is such a triplet present in array, then print the triplet and return true. Else return false.** Examples:

Input: array = {12, 3, 4, 1, 6, 9}, sum = 24;

Output: 12, 3, 9



The image shows a screenshot of a Java IDE with two windows. The left window is a command prompt titled 'C:\Windows\System32\cmd.exe' showing the execution of a Java program. The right window is the IDE editor showing the source code for 'TripletSumOfArray.java'.

**Command Prompt Output:**

```
C:\Windows\System32\cmd.exe
Enter size of array:6
-----
Enter array element:12 3 4 1 6 9
-----
Given Sum of triplet is:12 3 9
True
-----
C:\Users\Shree\Desktop\Practice_java_Codenera>
```

**Source Code (TripletSumOfArray.java):**

```
import java.util.*;
public class TripletSumOfArray
{
    public static void main(String[] ar)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter size of array:");
        int size = sc.nextInt();
        System.out.println("\n-----");
        int a[] = new int[size];
        System.out.print("Enter array element:");
        for(int i=0; i<a.length; i++)
        {
            a[i] = sc.nextInt();
        }
        System.out.println("\n-----");
        System.out.print("Given Sum of triplet is:");
        int t=24, c=0;
        for(int i=0; i<a.length; i++)
        {
            for(int j=i+1; j<a.length; j++)
            {
                for(int k=j+1; k<a.length; k++)
                {
                    int sum=0;
                    sum+=a[i]+a[j]+a[k];
                    if(sum==t)
                    {
                        System.out.println(a[i]+"
"+a[j]+" "+a[k]+" ");
                        c++;
                    }
                }
            }
        }
    }
}
```

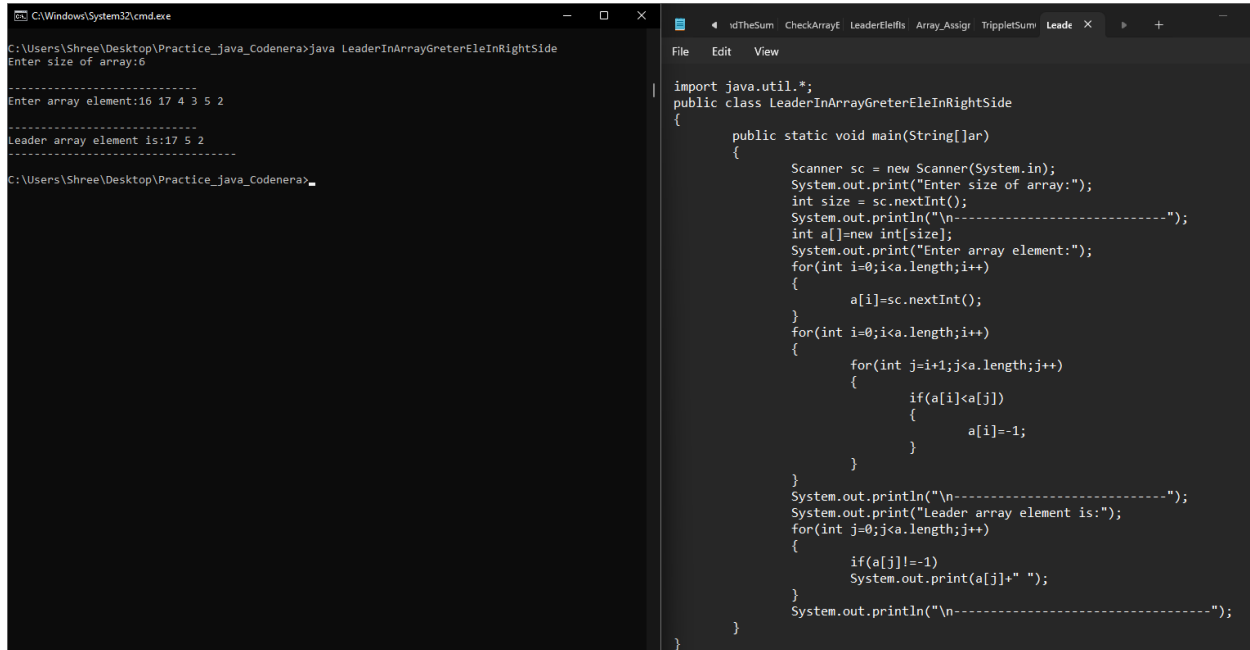
```
C:\Windows\System32\cmd.exe
Enter size of array:6
-----
Enter array element:12 3 4 1 6 9
-----
Given Sum of triplet is:12 3 9
True
-----
C:\Users\Shree\Desktop\Practice_java_Codenera>

File Edit View
a[i]=sc.nextInt();
}
System.out.println("\n-----");
System.out.print("Given Sum of triplet is:");
int t=24,c=0;
for(int i=0;i<a.length;i++)
{
    for(int j=i+1;j<a.length;j++)
    {
        for(int k=j+1;k<a.length;k++)
        {
            int sum=0;
            sum+=a[i]+a[j]+a[k];
            if(sum==t)
            {
                System.out.println(a[i]+"
"+a[j]+" "+a[k]+" ");
                c++;
            }
        }
    }
}
if(c==1)
{
    System.out.println("True");
    c++;
}
else
{
    System.out.println("False");
}
System.out.println("\n-----");
}
```

2. Write a Java program to print all the **LEADERS** in the array Note: An element is leader if it is greater than all the elements to its right side.

input: a[] = {16, 17, 4, 3, 5, 2},

output : 17 5 2.



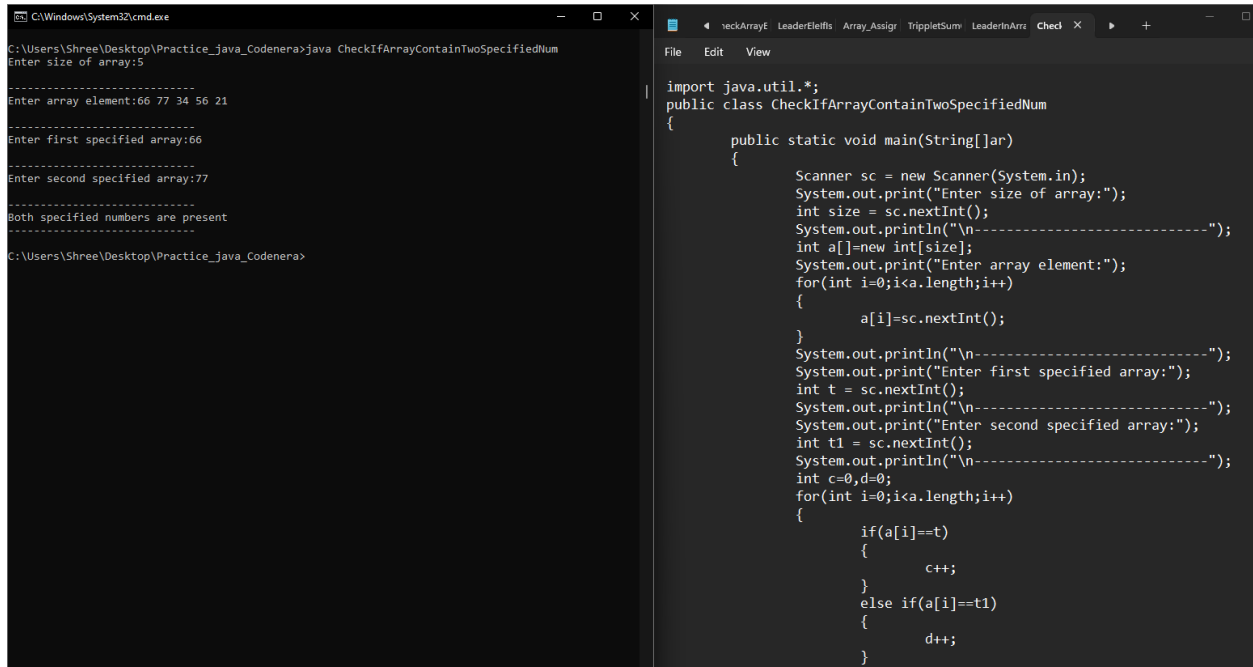
The screenshot shows a Java IDE on the right and a Windows command prompt on the left. The IDE contains the following Java code:

```
import java.util.*;
public class LeaderInArrayGreterEleInRightSide
{
    public static void main(String[]ar)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter size of array:");
        int size = sc.nextInt();
        System.out.println("\n-----");
        int a[]=new int[size];
        System.out.print("Enter array element:");
        for(int i=0;i<a.length;i++)
        {
            a[i]=sc.nextInt();
        }
        for(int i=0;i<a.length;i++)
        {
            for(int j=i+1;j<a.length;j++)
            {
                if(a[i]<a[j])
                {
                    a[i]=-1;
                }
            }
        }
        System.out.println("\n-----");
        System.out.print("Leader array element is:");
        for(int j=0;j<a.length;j++)
        {
            if(a[j]!=-1)
                System.out.print(a[j]+" ");
        }
        System.out.println("\n-----");
    }
}
```

The command prompt shows the execution of the program:

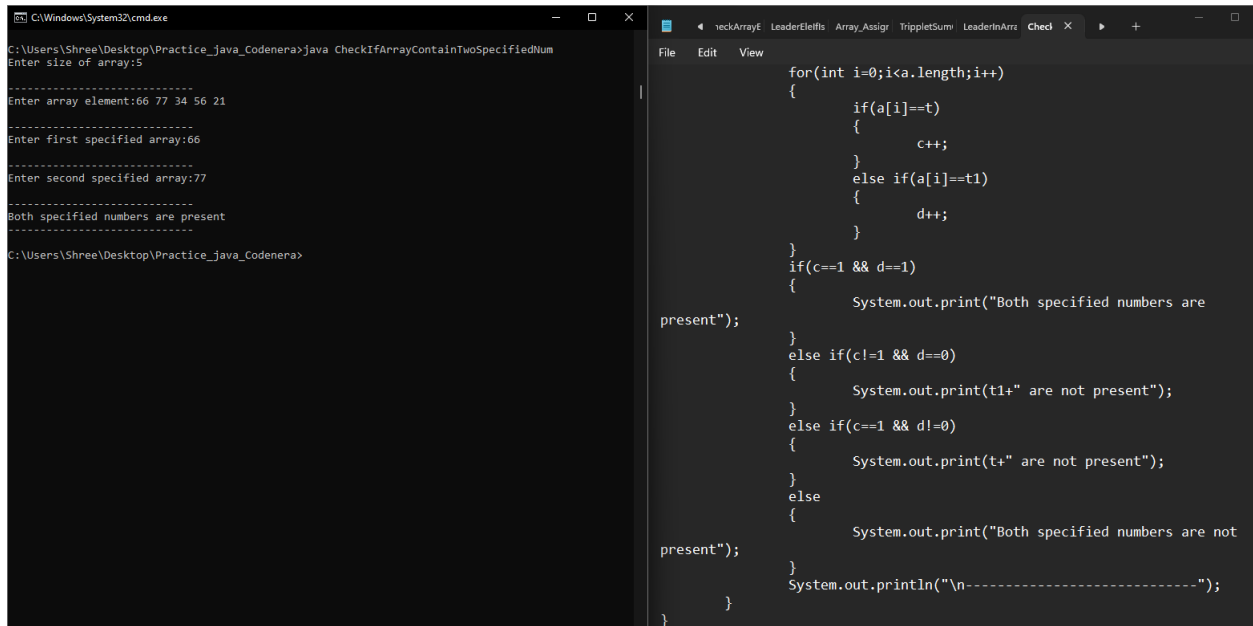
```
C:\Users\Shree\Desktop\Practice_java_Codenera>java LeaderInArrayGreterEleInRightSide
Enter size of array:6
-----
Enter array element:16 17 4 3 5 2
-----
Leader array element is:17 5 2
-----
C:\Users\Shree\Desktop\Practice_java_Codenera>
```

### 3. Write a Java program to check if an array of integers contains two specified elements 65 and 77.



```
C:\Windows\System32\cmd.exe
C:\Users\Shree\Desktop\Practice_java_Codenera>java CheckIfArrayContainTwoSpecifiedNum
Enter size of array:5
-----
Enter array element:66 77 34 56 21
-----
Enter first specified array:66
-----
Enter second specified array:77
-----
Both specified numbers are present
-----
C:\Users\Shree\Desktop\Practice_java_Codenera>
```

```
File Edit View
import java.util.*;
public class CheckIfArrayContainTwoSpecifiedNum
{
    public static void main(String[]ar)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter size of array:");
        int size = sc.nextInt();
        System.out.println("\n-----");
        int a[]=new int[size];
        System.out.print("Enter array element:");
        for(int i=0;i<a.length;i++)
        {
            a[i]=sc.nextInt();
        }
        System.out.println("\n-----");
        System.out.print("Enter first specified array:");
        int t = sc.nextInt();
        System.out.println("\n-----");
        System.out.print("Enter second specified array:");
        int t1 = sc.nextInt();
        System.out.println("\n-----");
        int c=0,d=0;
        for(int i=0;i<a.length;i++)
        {
            if(a[i]==t)
            {
                c++;
            }
            else if(a[i]==t1)
            {
                d++;
            }
        }
    }
}
```



```
C:\Windows\System32\cmd.exe
C:\Users\Shree\Desktop\Practice_java_Codenera>java CheckIfArrayContainTwoSpecifiedNum
Enter size of array:5
-----
Enter array element:66 77 34 56 21
-----
Enter first specified array:66
-----
Enter second specified array:77
-----
Both specified numbers are present
-----
C:\Users\Shree\Desktop\Practice_java_Codenera>
```

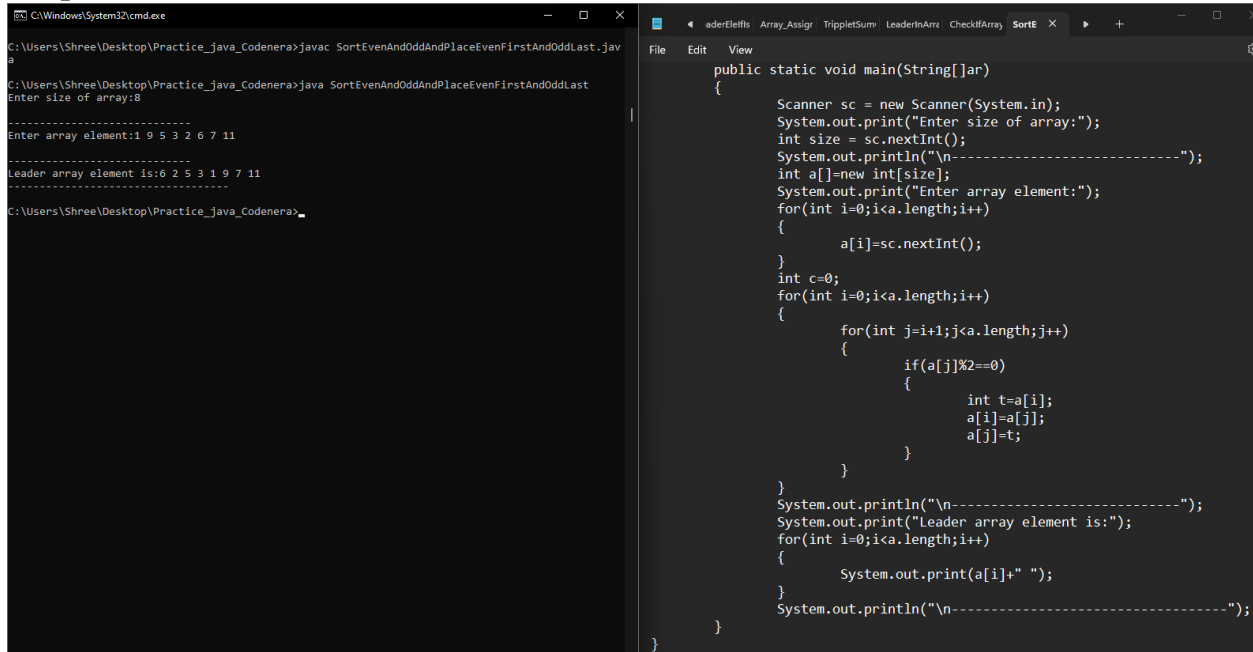
```
File Edit View
for(int i=0;i<a.length;i++)
{
    if(a[i]==t)
    {
        c++;
    }
    else if(a[i]==t1)
    {
        d++;
    }
}
if(c==1 && d==1)
{
    System.out.print("Both specified numbers are
present");
}
else if(c!=1 && d==0)
{
    System.out.print(t1+" are not present");
}
else if(c==1 && d!=0)
{
    System.out.print(t+" are not present");
}
else
{
    System.out.print("Both specified numbers are not
present");
}
System.out.println("\n-----");
}
```

4. Given an array `arr[]` of integers, segregate even and odd numbers in the array. Such that all the even numbers should be present first, and then the odd numbers.

Examples:

Input: `arr[] = 1 9 5 3 2 6 7 11`

Output: `2 6 5 3 1 9 7 11`



The image shows a screenshot of a Java IDE with two windows. The left window is a command prompt showing the execution of a Java program. The right window is the IDE editor showing the source code of the program.

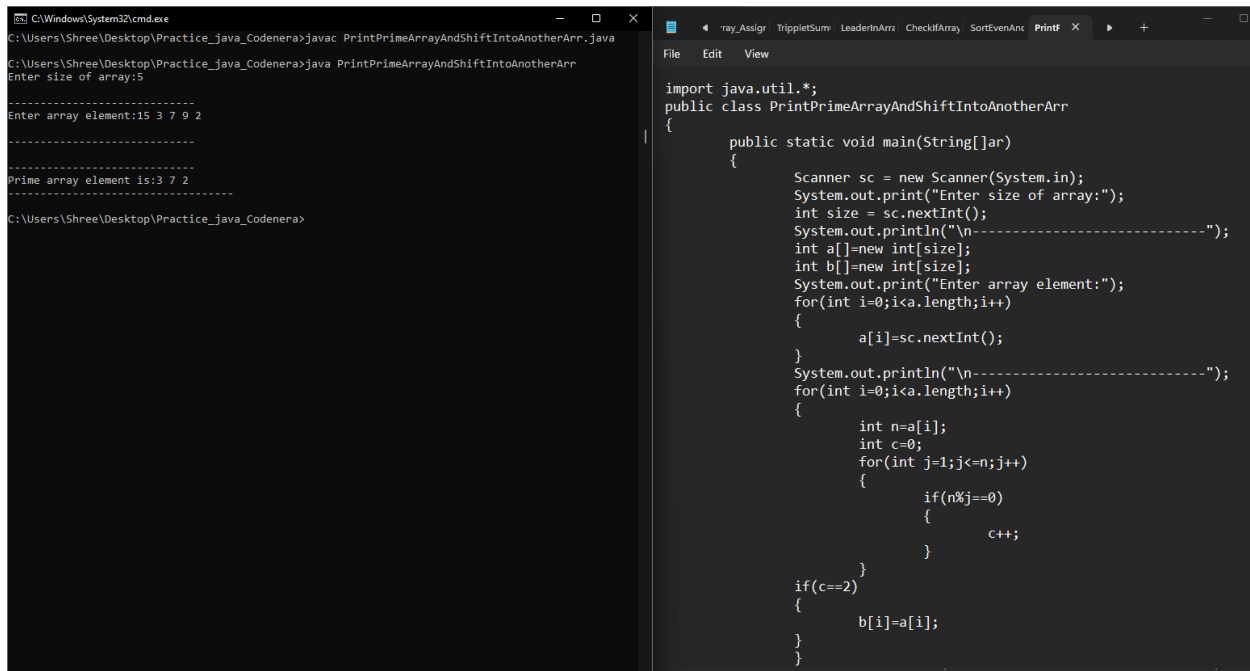
**Command Prompt Output:**

```
C:\Users\Shree\Desktop\Practice_java_Codenera>javac SortEvenAndOddAndPlaceEvenFirstAndOddLast.java
C:\Users\Shree\Desktop\Practice_java_Codenera>java SortEvenAndOddAndPlaceEvenFirstAndOddLast
Enter size of array:8
-----
Enter array element:1 9 5 3 2 6 7 11
-----
Leader array element is:2 6 5 3 1 9 7 11
-----
C:\Users\Shree\Desktop\Practice_java_Codenera>
```

**Source Code:**

```
public static void main(String[]ar)
{
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter size of array:");
    int size = sc.nextInt();
    System.out.println("\n-----");
    int a[]=new int[size];
    System.out.print("Enter array element:");
    for(int i=0;i<a.length;i++)
    {
        a[i]=sc.nextInt();
    }
    int c=0;
    for(int i=0;i<a.length;i++)
    {
        for(int j=i+1;j<a.length;j++)
        {
            if(a[j]%2==0)
            {
                int t=a[i];
                a[i]=a[j];
                a[j]=t;
            }
        }
    }
    System.out.println("\n-----");
    System.out.print("Leader array element is:");
    for(int i=0;i<a.length;i++)
    {
        System.out.print(a[i]+" ");
    }
    System.out.println("\n-----");
}
```

5. Write a java program to find prime number between an array of element and shift into another array and print that prime array.

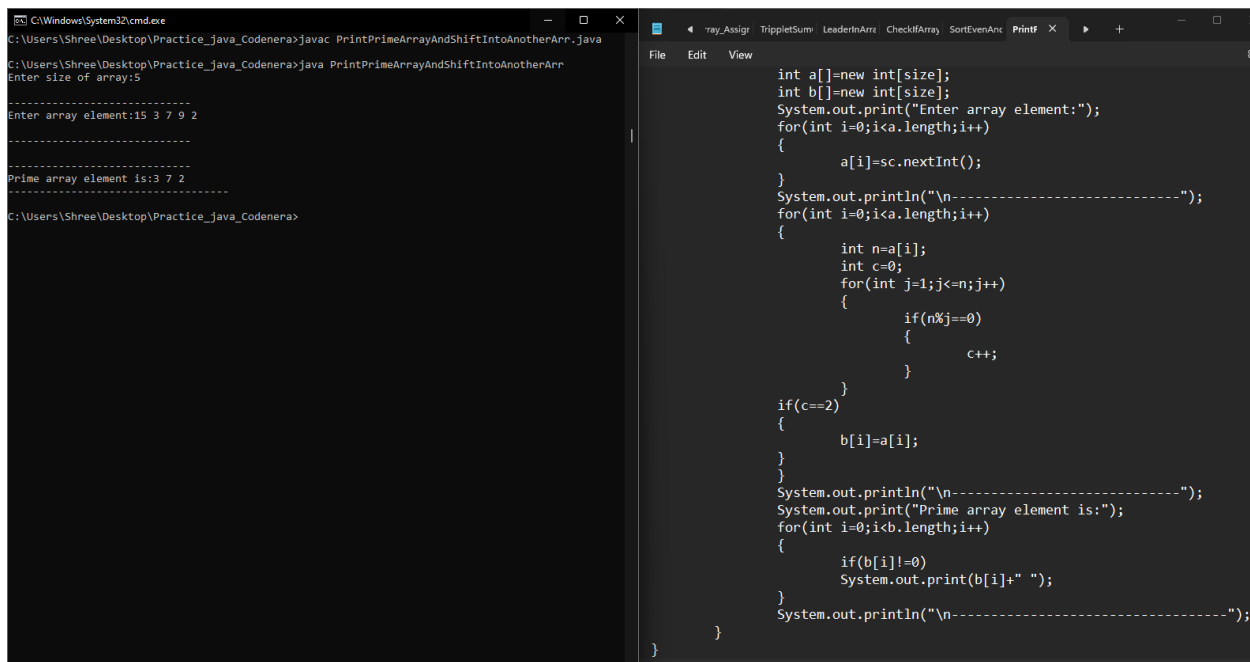


The screenshot shows a Java IDE on the right and a Windows command prompt on the left. The IDE contains the following code:

```
import java.util.*;
public class PrintPrimeArrayAndShiftIntoAnotherArr
{
    public static void main(String[]ar)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter size of array:");
        int size = sc.nextInt();
        System.out.println("\n-----");
        int a[]=new int[size];
        int b[]=new int[size];
        System.out.print("Enter array element:");
        for(int i=0;i<a.length;i++)
        {
            a[i]=sc.nextInt();
        }
        System.out.println("\n-----");
        for(int i=0;i<a.length;i++)
        {
            int n=a[i];
            int c=0;
            for(int j=1;j<=n;j++)
            {
                if(n%j==0)
                {
                    c++;
                }
            }
            if(c==2)
            {
                b[i]=a[i];
            }
        }
    }
}
```

The command prompt shows the execution of the program:

```
C:\Windows\System32\cmd.exe
C:\Users\Shree\Desktop\Practice_java_Codenera>javac PrintPrimeArrayAndShiftIntoAnotherArr.java
C:\Users\Shree\Desktop\Practice_java_Codenera>java PrintPrimeArrayAndShiftIntoAnotherArr
Enter size of array:5
-----
Enter array element:15 3 7 9 2
-----
Prime array element is:3 7 2
C:\Users\Shree\Desktop\Practice_java_Codenera>
```



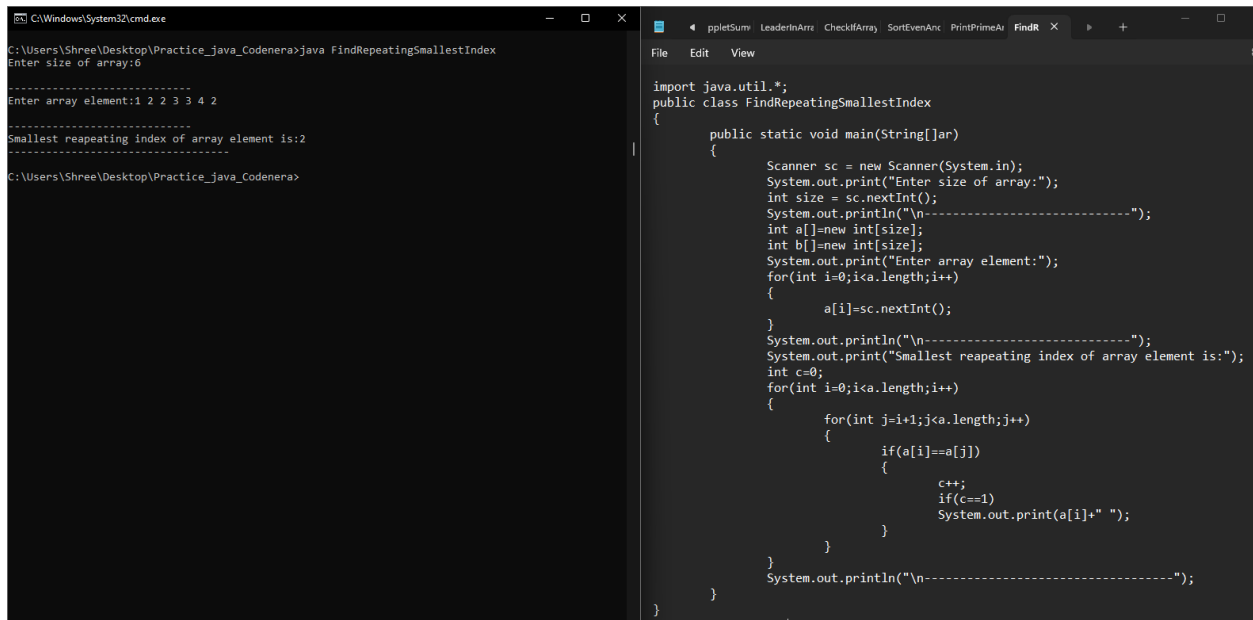
The screenshot shows a Java IDE on the right and a Windows command prompt on the left. The IDE contains the following code:

```
int a[]=new int[size];
int b[]=new int[size];
System.out.print("Enter array element:");
for(int i=0;i<a.length;i++)
{
    a[i]=sc.nextInt();
}
System.out.println("\n-----");
for(int i=0;i<a.length;i++)
{
    int n=a[i];
    int c=0;
    for(int j=1;j<=n;j++)
    {
        if(n%j==0)
        {
            c++;
        }
    }
    if(c==2)
    {
        b[i]=a[i];
    }
}
System.out.println("\n-----");
System.out.print("Prime array element is:");
for(int i=0;i<b.length;i++)
{
    if(b[i]!=0)
    System.out.print(b[i]+" ");
}
System.out.println("\n-----");
}
```

The command prompt shows the execution of the program:

```
C:\Windows\System32\cmd.exe
C:\Users\Shree\Desktop\Practice_java_Codenera>javac PrintPrimeArrayAndShiftIntoAnotherArr.java
C:\Users\Shree\Desktop\Practice_java_Codenera>java PrintPrimeArrayAndShiftIntoAnotherArr
Enter size of array:5
-----
Enter array element:15 3 7 9 2
-----
Prime array element is:3 7 2
C:\Users\Shree\Desktop\Practice_java_Codenera>
```

**6. 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.**



```
C:\Windows\System32\cmd.exe
C:\Users\Shree\Desktop\Practice_java_Codenera>java FindRepeatingSmallestIndex
Enter size of array:6
-----
Enter array element:1 2 2 3 3 4 2
-----
Smallest repeating index of array element is:2
-----
C:\Users\Shree\Desktop\Practice_java_Codenera>
```

```
File Edit View
import java.util.*;
public class FindRepeatingSmallestIndex
{
    public static void main(String[] ar)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter size of array:");
        int size = sc.nextInt();
        System.out.println("\n-----");
        int a[]=new int[size];
        int b[]=new int[size];
        System.out.print("Enter array element:");
        for(int i=0;i<a.length;i++)
        {
            a[i]=sc.nextInt();
        }
        System.out.println("\n-----");
        System.out.print("Smallest repeating index of array element is:");
        int c=0;
        for(int i=0;i<a.length;i++)
        {
            for(int j=i+1;j<a.length;j++)
            {
                if(a[i]==a[j])
                {
                    c++;
                    if(c==1)
                        System.out.print(a[i]+" ");
                }
            }
        }
        System.out.println("\n-----");
    }
}
```

7. .Given two unsorted arrays that represent two sets (elements in every array are distinct), find the union and intersection of two arrays.

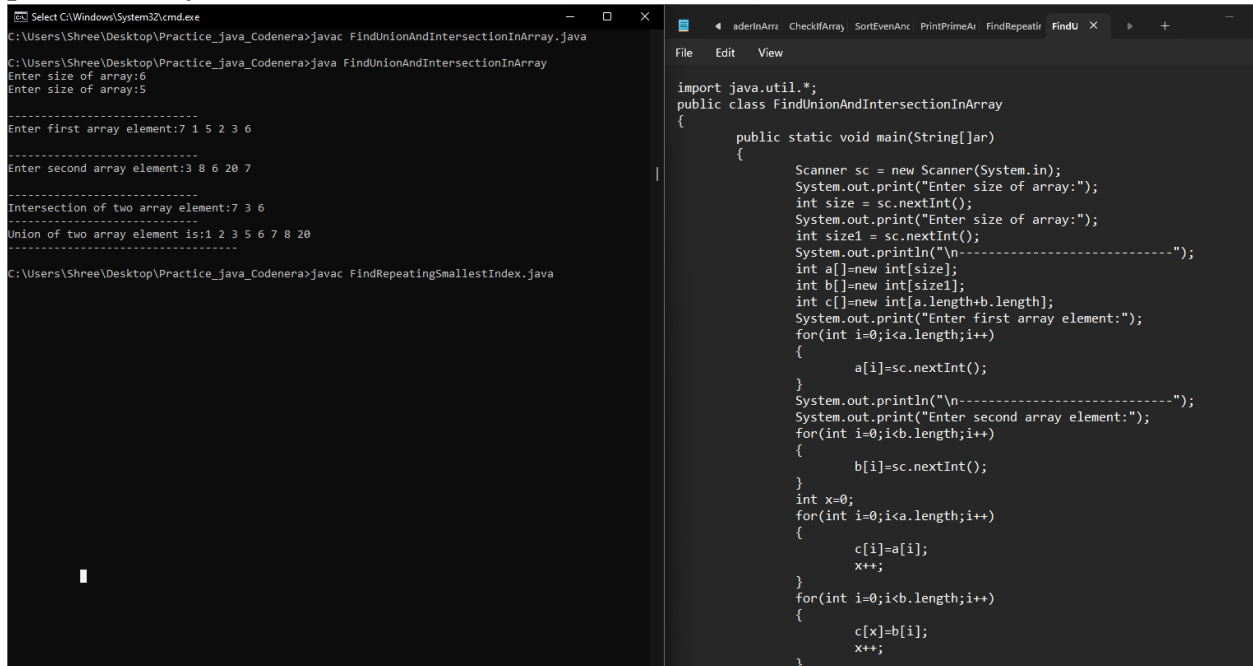
Example:

arr1[] = {7, 1, 5, 2, 3, 6}

arr2[] = {3, 8, 6, 20, 7}

Then your program should print Union as {1, 2, 3, 5, 6, 7, 8, 20}

And Intersection as {3, 6, 7}. Note that the elements of union and intersection can be printed in any order.



The screenshot displays a Java IDE with two windows. The left window shows the command prompt output for running the program, and the right window shows the source code of the program.

**Command Prompt Output:**

```
C:\Users\Shree\Desktop\Practice_java_Codenera>javac FindUnionAndIntersectionInArray.java
C:\Users\Shree\Desktop\Practice_java_Codenera>java FindUnionAndIntersectionInArray
Enter size of array:6
Enter size of array:5
-----
Enter first array element:7 1 5 2 3 6
-----
Enter second array element:3 8 6 20 7
-----
Intersection of two array element:7 3 6
-----
Union of two array element is:1 2 3 5 6 7 8 20
-----
C:\Users\Shree\Desktop\Practice_java_Codenera>javac FindRepeatingSmallestIndex.java
```

**Source Code:**

```
import java.util.*;
public class FindUnionAndIntersectionInArray
{
    public static void main(String[] ar)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter size of array:");
        int size = sc.nextInt();
        System.out.print("Enter size of array:");
        int size1 = sc.nextInt();
        System.out.println("\n-----");
        int a[] = new int[size];
        int b[] = new int[size1];
        int c[] = new int[a.length+b.length];
        System.out.print("Enter first array element:");
        for(int i=0; i<a.length; i++)
        {
            a[i] = sc.nextInt();
        }
        System.out.println("\n-----");
        System.out.print("Enter second array element:");
        for(int i=0; i<b.length; i++)
        {
            b[i] = sc.nextInt();
        }
        int x=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++;
        }
    }
}
```



Select C:\Windows\System32\cmd.exe

C:\Users\Shree\Desktop\Practice\_java\_Codenera>javac FindUnionAndIntersectionInArray.java  
C:\Users\Shree\Desktop\Practice\_java\_Codenera>java FindUnionAndIntersectionInArray  
Enter size of array:6  
Enter size of array:5  
-----  
Enter first array element:7 1 5 2 3 6  
-----  
Enter second array element:3 8 6 20 7  
-----  
Intersection of two array element:7 3 6  
-----  
Union of two array element is:1 2 3 5 6 7 8 20  
-----  
C:\Users\Shree\Desktop\Practice\_java\_Codenera>javac FindRepeatingSmallestIndex.java

aderInArry CheckIfArray SortEvenAnc PrintPrimeAi FindRepeati FindU X

File Edit View  
System.out.println("\n-----");  
System.out.print("Intersection of two array element:");  
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]+" ");  
            }  
            c[j]=-1;  
        }  
    }  
}  
for(int i=0;i<c.length;i++)  
{  
    for(int j=i+1;j<c.length;j++)  
    {  
        if(c[i]>c[j])  
        {  
            int t=c[i];  
            c[i]=c[j];  
            c[j]=t;  
        }  
    }  
}  
System.out.println("\n-----");  
System.out.print("Union of two array element is:");  
for(int i=0;i<c.length;i++)  
{  
    if(c[i]!=-1)  
        System.out.print(c[i]+" ");  
}  
System.out.println("\n-----");  
}

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

Examples:

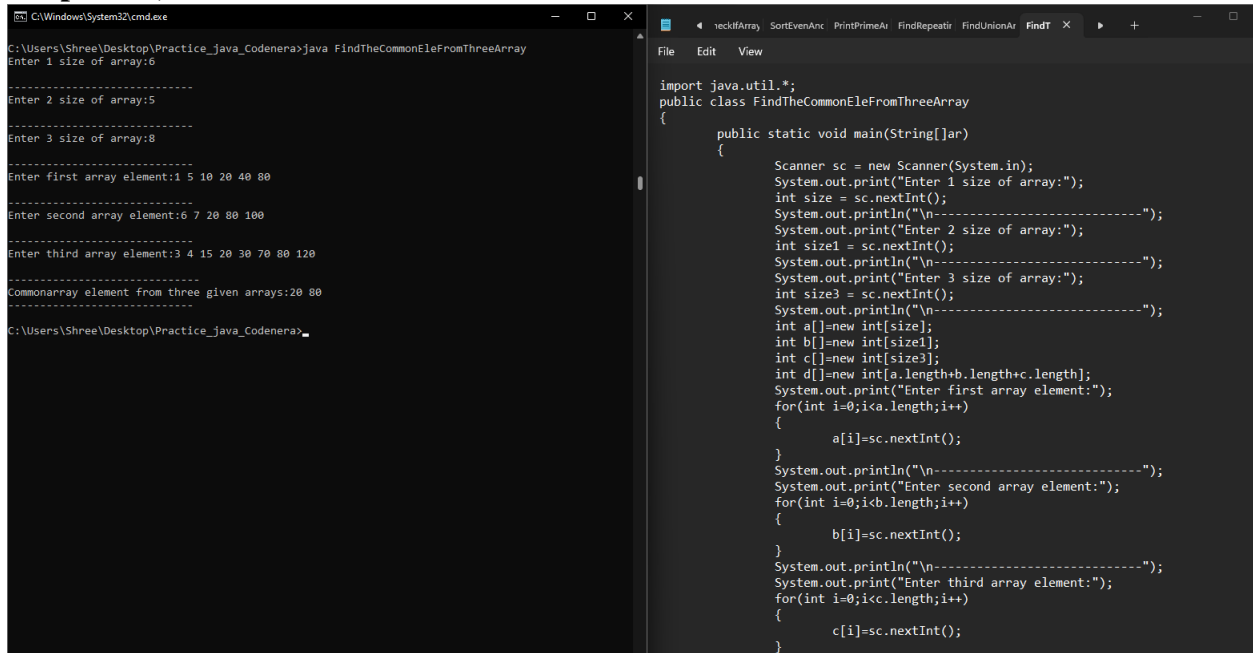
Input:

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



The screenshot displays a Java IDE with two windows. The left window is a command prompt running the program, and the right window shows the source code.

**Command Prompt Output:**

```
C:\Users\Shree\Desktop\Practice_java_Codenera>java FindTheCommonEleFromThreeArray
Enter 1 size of array:6
-----
Enter 2 size of array:5
-----
Enter 3 size of array:8
-----
Enter first array element:1 5 10 20 40 80
-----
Enter second array element:6 7 20 80 100
-----
Enter third array element:3 4 15 20 30 70 80 120
-----
Commonarray element from three given arrays:20 80
-----
C:\Users\Shree\Desktop\Practice_java_Codenera>
```

**Source Code:**

```
import java.util.*;
public class FindTheCommonEleFromThreeArray
{
    public static void main(String[]ar)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter 1 size of array:");
        int size = sc.nextInt();
        System.out.println("\n-----");
        System.out.print("Enter 2 size of array:");
        int size1 = sc.nextInt();
        System.out.println("\n-----");
        System.out.print("Enter 3 size of array:");
        int size3 = sc.nextInt();
        System.out.println("\n-----");
        int a[]=new int[size];
        int b[]=new int[size1];
        int c[]=new int[size3];
        int d[]=new int[a.length+b.length+c.length];
        System.out.print("Enter first array element:");
        for(int i=0;i<a.length;i++)
        {
            a[i]=sc.nextInt();
        }
        System.out.println("\n-----");
        System.out.print("Enter second array element:");
        for(int i=0;i<b.length;i++)
        {
            b[i]=sc.nextInt();
        }
        System.out.println("\n-----");
        System.out.print("Enter third array element:");
        for(int i=0;i<c.length;i++)
        {
            c[i]=sc.nextInt();
        }
    }
}
```

```
C:\Windows\System32\cmd.exe
C:\Users\Shree\Desktop\Practice_java_Codenera>java FindTheCommonEleFromThreeArray
Enter 1 size of array:6
-----
Enter 2 size of array:5
-----
Enter 3 size of array:8
-----
Enter first array element:1 5 10 20 40 80
-----
Enter second array element:6 7 20 80 100
-----
Enter third array element:3 4 15 20 30 70 80 120
-----
Commonarray element from three given arrays:20 80
-----
C:\Users\Shree\Desktop\Practice_java_Codenera>
```

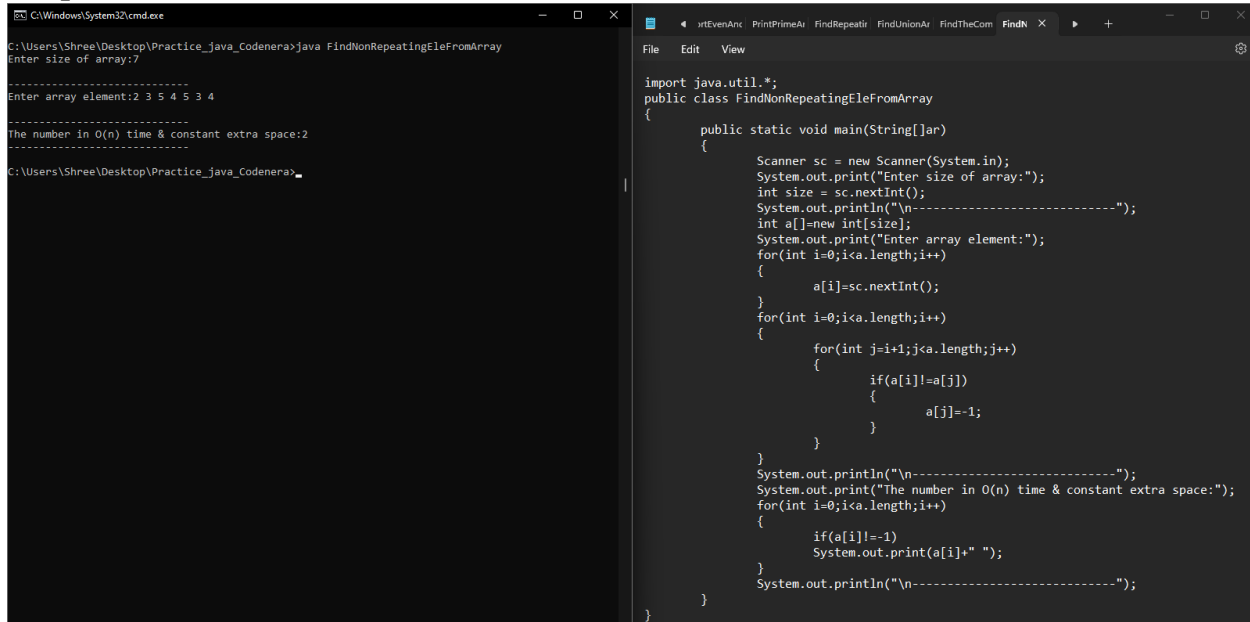
```
File Edit View
FindT x
}
int x=0;
for(int i=0;i<a.length;i++)
{
    d[i]=a[i];
    x++;
}
for(int i=0;i<b.length;i++)
{
    d[x]=b[i];
    x++;
}
for(int i=0;i<c.length;i++)
{
    d[x]=c[i];
    x++;
}
System.out.println("\n-----");
System.out.print("Commonarray element from three given arrays:");
for(int i=0;i<d.length;i++)
{
    for(int j=i+1;j<d.length;j++)
    {
        if(d[i]==d[j])
        {
            if(d[i]!=-1 && d[j]!=-1)
            {
                System.out.print(d[i]+" ");
            }
            d[j]=-1;
            d[i]=-1;
        }
    }
}
System.out.println("\n-----");
}
```

**9. Given an array of integers. All numbers occur twice except one number which occurs once. Find the number in  $O(n)$  time & constant extra space.**

**Example :**

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

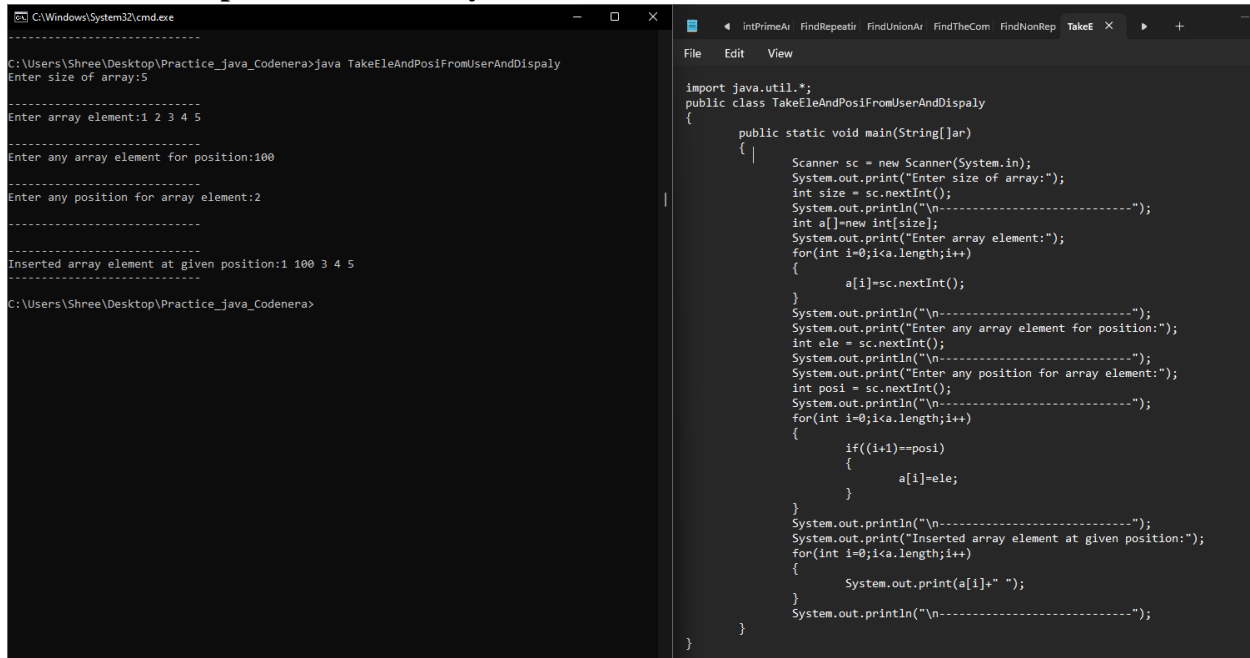
**Output:** 2



```
C:\Windows\System32\cmd.exe
C:\Users\Shree\Desktop\Practice_java_Codenera>java FindNonRepeatingEleFromArray
Enter size of array:7
-----
Enter array element:2 3 5 4 5 3 4
-----
The number in O(n) time & constant extra space:2
-----
C:\Users\Shree\Desktop\Practice_java_Codenera>

import java.util.*;
public class FindNonRepeatingEleFromArray
{
    public static void main(String[]ar)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter size of array:");
        int size = sc.nextInt();
        System.out.println("\n-----");
        int a[]=new int[size];
        System.out.print("Enter array element:");
        for(int i=0;i<a.length;i++)
        {
            a[i]=sc.nextInt();
        }
        for(int i=0;i<a.length;i++)
        {
            for(int j=i+1;j<a.length;j++)
            {
                if(a[i]!=a[j])
                {
                    a[j]=-1;
                }
            }
        }
        System.out.println("\n-----");
        System.out.print("The number in O(n) time & constant extra space:");
        for(int i=0;i<a.length;i++)
        {
            if(a[i]==-1)
                System.out.print(a[i]+" ");
        }
        System.out.println("\n-----");
    }
}
```

## 10. Write a program input an array now insert any element at any position, Element and position is taken by user.



The image shows a Java IDE on the right and a Windows command prompt on the left. The IDE displays the source code for a class named `TakeEleAndPosiFromUserAndDispaly`. The code uses a `Scanner` to take input for the array size, elements, and the position to insert a new element. It then shifts the existing elements one position to the right and inserts the new element at the specified position. The command prompt shows the execution of the program, where the user enters an array size of 5, elements 1 2 3 4 5, and a position of 2. The output shows the array after insertion: 1 100 3 4 5.

```
C:\Windows\System32\cmd.exe
C:\Users\Shree\Desktop\Practice_java_Codenera>java TakeEleAndPosiFromUserAndDispaly
Enter size of array:5
Enter array element:1 2 3 4 5
Enter any array element for position:100
Enter any position for array element:2
Inserted array element at given position:1 100 3 4 5
C:\Users\Shree\Desktop\Practice_java_Codenera>
```

```
import java.util.*;
public class TakeEleAndPosiFromUserAndDispaly
{
    public static void main(String[]ar)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter size of array:");
        int size = sc.nextInt();
        System.out.println("\n-----");
        int a[]=new int[size];
        System.out.print("Enter array element:");
        for(int i=0;i<a.length;i++)
        {
            a[i]=sc.nextInt();
        }
        System.out.println("\n-----");
        System.out.print("Enter any array element for position:");
        int ele = sc.nextInt();
        System.out.println("\n-----");
        System.out.print("Enter any position for array element:");
        int posi = sc.nextInt();
        System.out.println("\n-----");
        for(int i=0;i<a.length;i++)
        {
            if((i+1)==posi)
            {
                a[i]=ele;
            }
        }
        System.out.println("\n-----");
        System.out.print("Inserted array element at given position:");
        for(int i=0;i<a.length;i++)
        {
            System.out.print(a[i]+" ");
        }
        System.out.println("\n-----");
    }
}
```

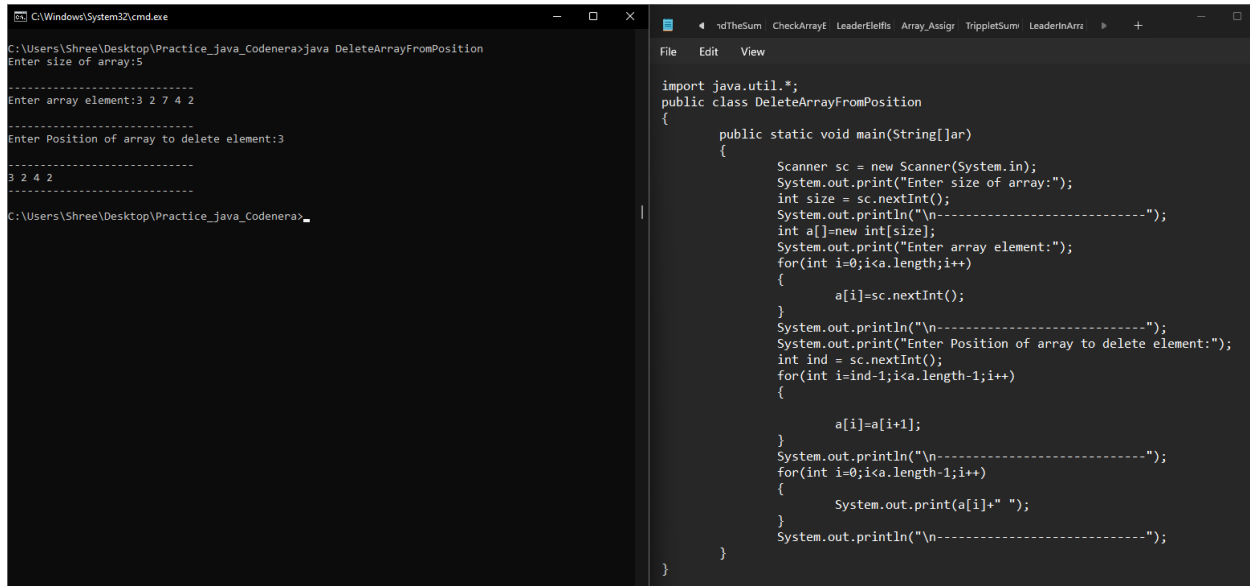
**11. Given a sorted array of  $n$  distinct integers where each integer is in the range from 0 to  $m-1$  and  $m > n$ . Find the smallest number that is missing from the array.**

**Examples:**

**Input: {0, 1, 2, 6, 9},  $n = 5$ ,  $m = 10$**

**Output: 3**

**12. Write a program input an array now delete element from array, position is taken from user.**



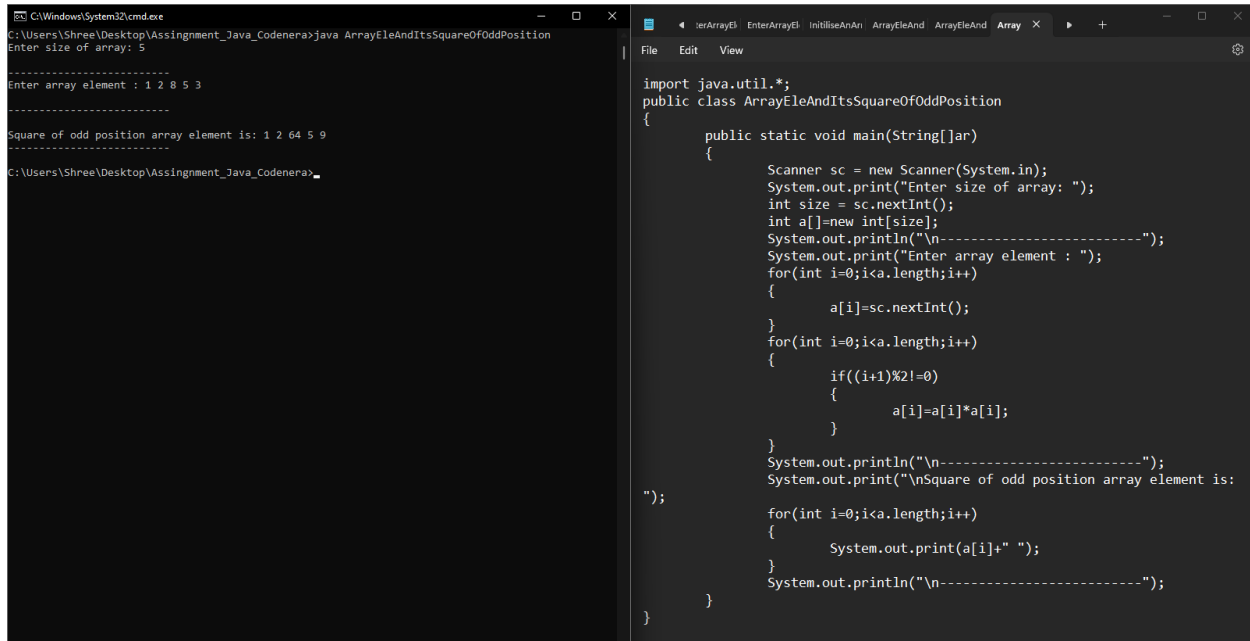
The image shows a Java IDE with a file explorer at the top displaying a project structure with files like 'ndTheSum', 'CheckArrayE', 'LeaderEleffis', 'Array\_Assign', 'TrippletSum', 'LeaderInArrz', and others. The main editor window contains the following Java code:

```
import java.util.*;
public class DeleteArrayFromPosition
{
    public static void main(String[]ar)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter size of array:");
        int size = sc.nextInt();
        System.out.println("\n\n-----");
        int a[]=new int[size];
        System.out.print("Enter array element:");
        for(int i=0;i<a.length;i++)
        {
            a[i]=sc.nextInt();
        }
        System.out.println("\n\n-----");
        System.out.print("Enter Position of array to delete element:");
        int ind = sc.nextInt();
        for(int i=ind-1;i<a.length-1;i++)
        {
            a[i]=a[i+1];
        }
        System.out.println("\n\n-----");
        for(int i=0;i<a.length-1;i++)
        {
            System.out.print(a[i]+" ");
        }
        System.out.println("\n\n-----");
    }
}
```

The terminal window on the left shows the execution of the program:

```
C:\Windows\System32\cmd.exe
C:\Users\Shree\Desktop\Practice_java_Codenera>java DeleteArrayFromPosition
Enter size of array:5
-----
Enter array element:3 2 7 4 2
-----
Enter Position of array to delete element:3
3 2 4 2
-----
C:\Users\Shree\Desktop\Practice_java_Codenera>
```

**13. Write a program enter an array and print the square of the element which is present at odd position.**



The image shows a screenshot of a Java program being executed in a command prompt and its source code in an IDE. The command prompt on the left shows the execution of the program, where the user enters the size of the array (5) and the array elements (1 2 8 5 3). The program outputs the square of the elements at odd positions (1, 8, 5), which are 1, 64, and 25, resulting in the output: 1 2 64 5 9.

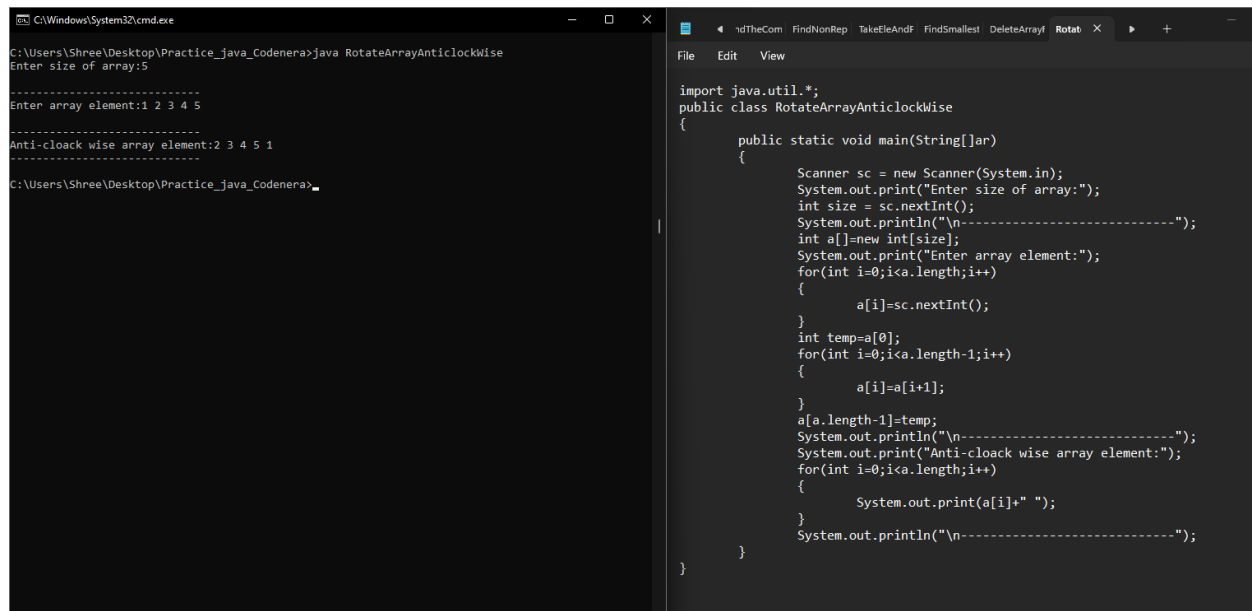
```
C:\Windows\System32\cmd.exe
C:\Users\Shree\Desktop\Assingment_Java_Codenera>java ArrayEleAndItsSquareOfOddPosition
Enter size of array: 5
-----
Enter array element : 1 2 8 5 3
-----
Square of odd position array element is: 1 2 64 5 9
-----
C:\Users\Shree\Desktop\Assingment_Java_Codenera>
```

The IDE on the right shows the source code of the program, which is a Java class named `ArrayEleAndItsSquareOfOddPosition`. The code uses a `Scanner` to take input from the user, calculates the square of elements at odd positions, and prints the result.

```
import java.util.*;
public class ArrayEleAndItsSquareOfOddPosition
{
    public static void main(String[] ar)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter size of array: ");
        int size = sc.nextInt();
        int a[]=new int[size];
        System.out.println("\n-----");
        System.out.print("Enter array element : ");
        for(int i=0;i<a.length;i++)
        {
            a[i]=sc.nextInt();
        }
        for(int i=0;i<a.length;i++)
        {
            if((i+1)%2!=0)
            {
                a[i]=a[i]*a[i];
            }
        }
        System.out.println("\n-----");
        System.out.print("\nSquare of odd position array element is: ");
        for(int i=0;i<a.length;i++)
        {
            System.out.print(a[i]+" ");
        }
        System.out.println("\n-----");
    }
}
```



#### 14. Write a program input an array and rotate it in anti-clock wise by any no given by user.



The image shows a screenshot of a Windows command prompt and a Java IDE. The command prompt on the left shows the execution of a Java program. The user enters the size of the array as 5, then the array elements 1 2 3 4 5, and finally the rotation value 2. The output shows the array after anti-clockwise rotation by 2 positions, resulting in 2 3 4 5 1. The Java IDE on the right shows the source code for the `RotateArrayAnticlockWise` class. The code uses a Scanner to take input, creates an array, and implements a rotation algorithm by shifting each element to the next index and placing the first element at the end.

```
C:\Windows\System32\cmd.exe
C:\Users\Shree\Desktop\Practice_java_Codenera>java RotateArrayAnticlockWise
Enter size of array:5
-----
Enter array element:1 2 3 4 5
-----
Anti-cloack wise array element:2 3 4 5 1
-----
C:\Users\Shree\Desktop\Practice_java_Codenera>_

import java.util.*;
public class RotateArrayAnticlockWise
{
    public static void main(String[] ar)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter size of array:");
        int size = sc.nextInt();
        System.out.println("\n-----");
        int a[]=new int[size];
        System.out.print("Enter array element:");
        for(int i=0;i<a.length;i++)
        {
            a[i]=sc.nextInt();
        }
        int temp=a[0];
        for(int i=0;i<a.length-1;i++)
        {
            a[i]=a[i+1];
        }
        a[a.length-1]=temp;
        System.out.println("\n-----");
        System.out.print("Anti-cloack wise array element:");
        for(int i=0;i<a.length;i++)
        {
            System.out.print(a[i]+" ");
        }
        System.out.println("\n-----");
    }
}
```

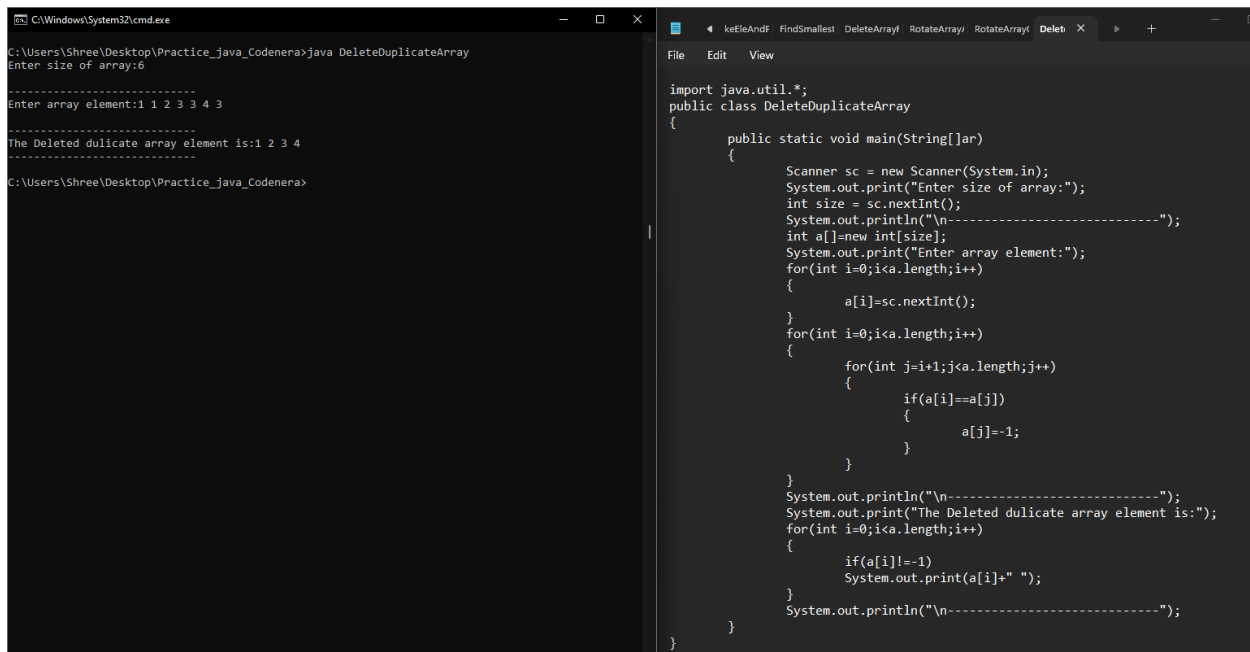
## 15. Write a program input an array and rotate it in clock wise by any no given by user.

```
C:\Windows\System32\cmd.exe
C:\Users\Shree\Desktop\Practice_java_Codenera>java RotateArrayClockWise
Enter size of array:5
-----
Enter array element:1 2 3 4 5
-----
Clock wise array element:5 1 2 3 4
-----
C:\Users\Shree\Desktop\Practice_java_Codenera>
```

```
File Edit View

import java.util.*;
public class RotateArrayClockWise
{
    public static void main(String[]ar)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter size of array:");
        int size = sc.nextInt();
        System.out.println("\n-----");
        int a[]=new int[size];
        System.out.print("Enter array element:");
        for(int i=0;i<a.length;i++)
        {
            a[i]=sc.nextInt();
        }
        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.println("\n-----");
        System.out.print("Clock wise array element:");
        for(int i=0;i<a.length;i++)
        {
            System.out.print(a[i]+" ");
        }
        System.out.println("\n-----");
    }
}
```

## 16. Write a program input an array and delete all duplicate element from array.



The image shows a Java program designed to remove duplicate elements from an array. The program is implemented in a class named `DeleteDuplicateArray`. It uses a `Scanner` to take user input for the array size and its elements. The logic involves a nested loop where each element is compared with all subsequent elements. If a duplicate is found, the element at the later index is set to `-1`. Finally, the array is printed, skipping any `-1` values.

```
import java.util.*;
public class DeleteDuplicateArray
{
    public static void main(String[] ar)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter size of array:");
        int size = sc.nextInt();
        System.out.println("\n-----");
        int a[]=new int[size];
        System.out.print("Enter array element:");
        for(int i=0;i<a.length;i++)
        {
            a[i]=sc.nextInt();
        }
        for(int i=0;i<a.length;i++)
        {
            for(int j=i+1;j<a.length;j++)
            {
                if(a[i]==a[j])
                {
                    a[j]=-1;
                }
            }
        }
        System.out.println("\n-----");
        System.out.print("The Deleted duplicate array element is:");
        for(int i=0;i<a.length;i++)
        {
            if(a[i]!=-1)
                System.out.print(a[i]+" ");
        }
        System.out.println("\n-----");
    }
}
```

The execution output in the command prompt is as follows:

```
C:\Users\Shree\Desktop\Practice_java_Codenera>java DeleteDuplicateArray
Enter size of array:6
-----
Enter array element:1 1 2 3 3 4 3
-----
The Deleted duplicate array element is:1 2 3 4
-----
C:\Users\Shree\Desktop\Practice_java_Codenera>
```