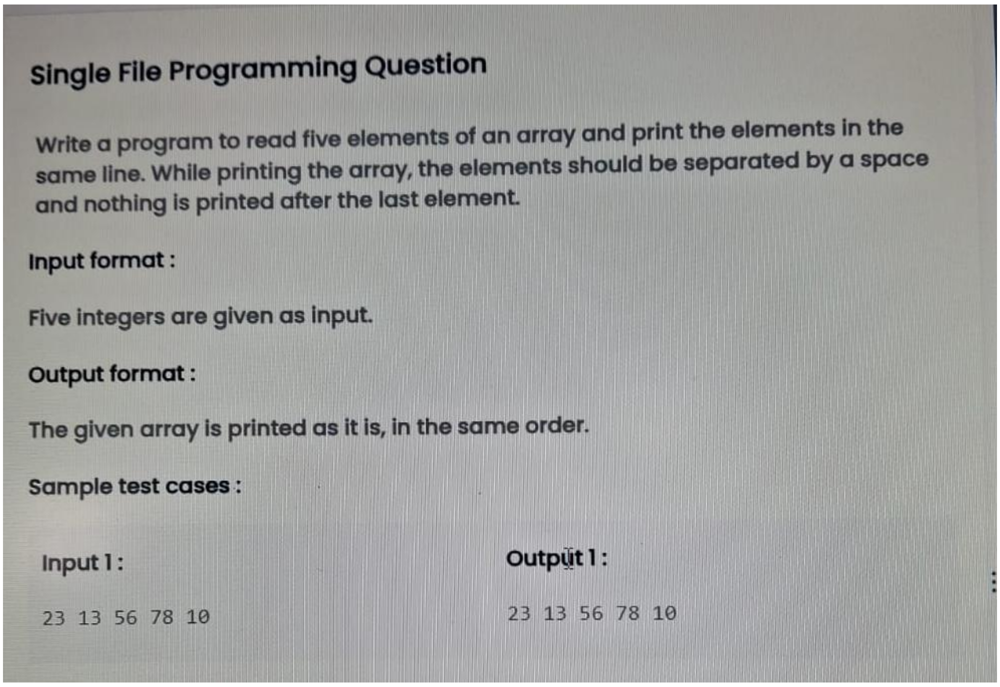
1. Data structure 1_Arrays_Chapter 1

TASK1:



Single File Programming Question

Write a program to read five elements of an array and print the elements in the same line. While printing the array, the elements should be separated by a space and nothing is printed after the last element.

Input format :

Five integers are given as input.

Output format :

The given array is printed as it is, in the same order.

Sample test cases :

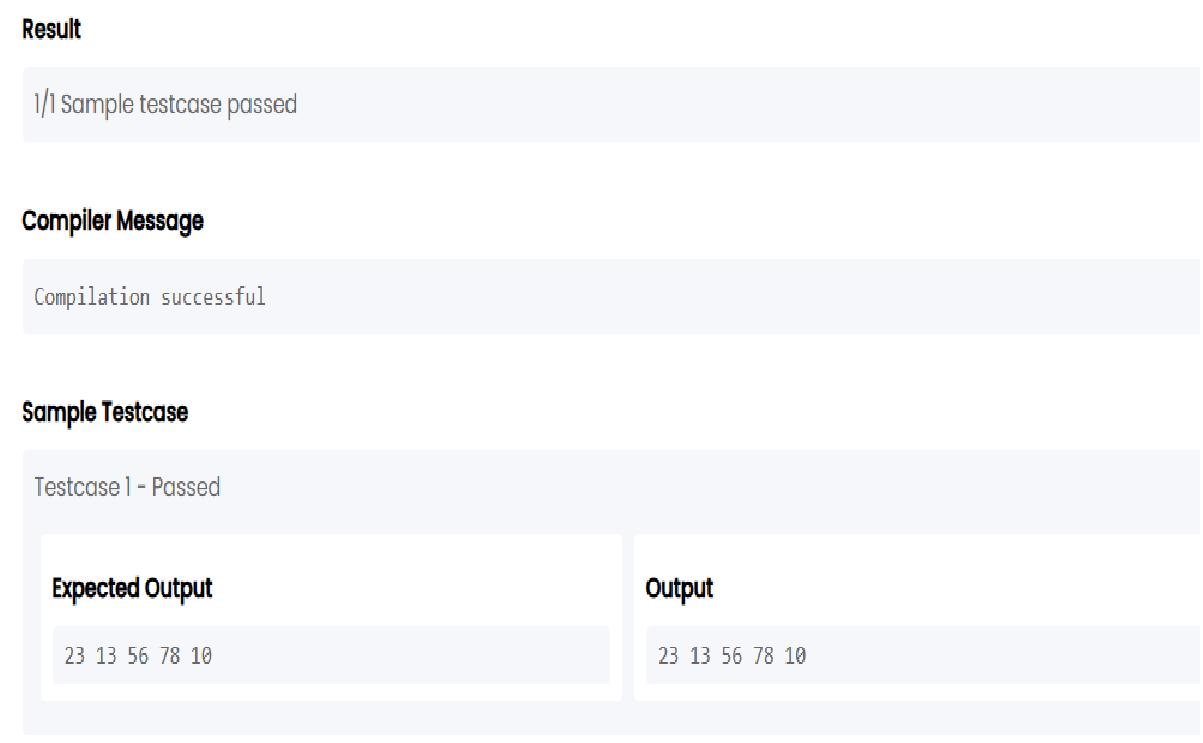
Input 1 :

```
23 13 56 78 10
```

Output 1 :

```
23 13 56 78 10
```

```
import java.util.*;
public class Main {
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int[] a = new int[5];
    for(int i=0;i<5;i++)
      a[i] = sc.nextInt();
    for(int i=0;i<5;i++){
      System.out.print(a[i]);
      if(i<4) System.out.print(" ");
    }
  }
}
```



Result

1/1 Sample testcase passed

Compiler Message

```
Compilation successful
```

Sample Testcase

Testcase 1 - Passed

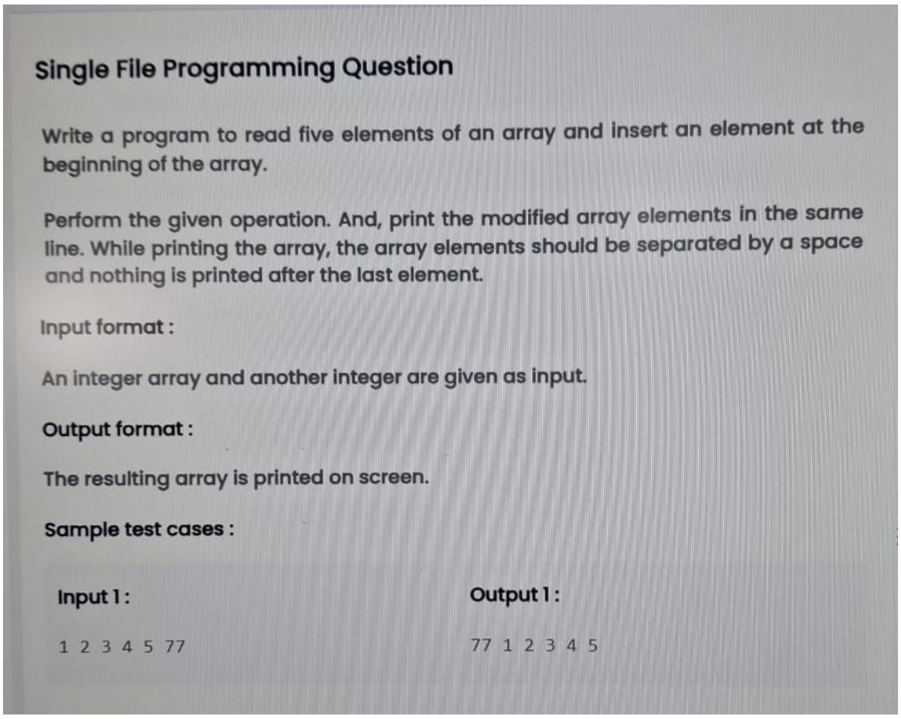
Expected Output

```
23 13 56 78 10
```

Output

```
23 13 56 78 10
```

TASK2:



Single File Programming Question

Write a program to read five elements of an array and insert an element at the beginning of the array.

Perform the given operation. And, print the modified array elements in the same line. While printing the array, the array elements should be separated by a space and nothing is printed after the last element.

Input format :

An integer array and another integer are given as input.

Output format :

The resulting array is printed on screen.

Sample test cases :

Input 1 :

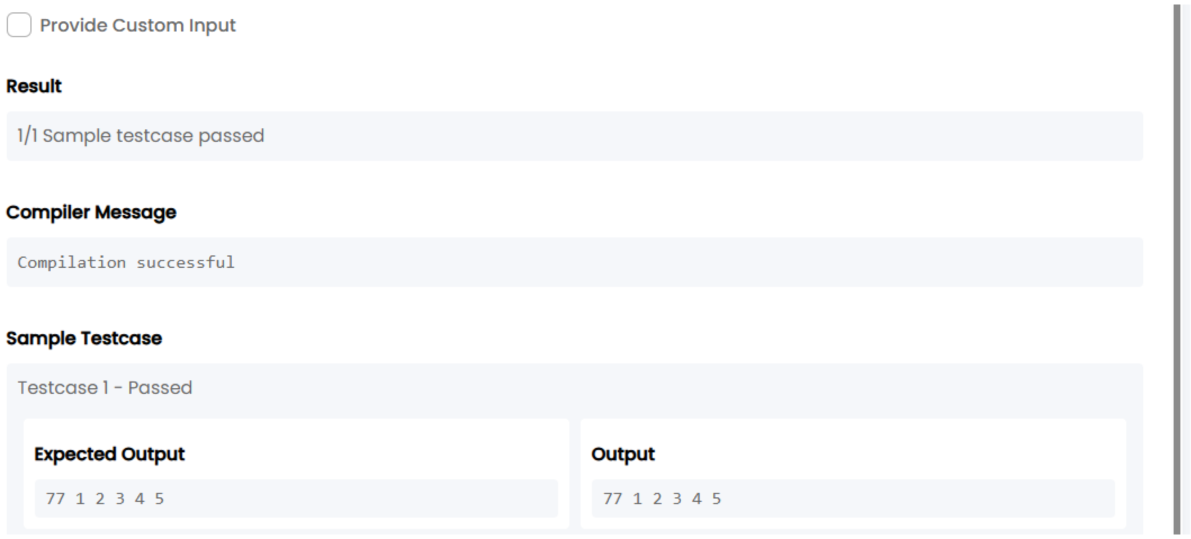
```
1 2 3 4 5 77
```

Output 1 :

```
77 1 2 3 4 5
```

```
import java.util.*;
public class Main {
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int[] a = new int[5];
```

```
      for(int i=0;i<5;i++)
         a[i] = sc.nextInt();
          int x = sc.nextInt();
          System.out.print(x + " ");
      for(int i=0;i<5;i++){
         System.out.print(a[i]);
         if(i<4) System.out.print(" ");
      }
   }
}
```



□ Provide Custom Input

Result

1/1 Sample testcase passed

Compiler Message

```
Compilation successful
```

Sample Testcase

Testcase 1 - Passed

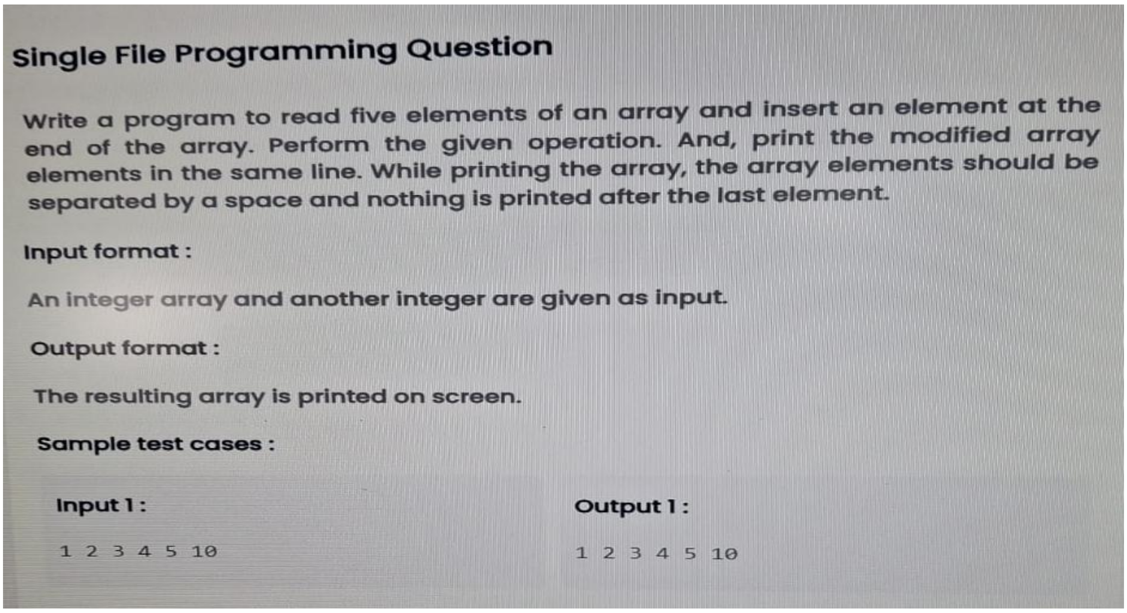
Expected Output

```
77 1 2 3 4 5
```

Output

```
77 1 2 3 4 5
```

TASK3:



Single File Programming Question

Write a program to read five elements of an array and insert an element at the end of the array. Perform the given operation. And, print the modified array elements in the same line. While printing the array, the array elements should be separated by a space and nothing is printed after the last element.

Input format :

An integer array and another integer are given as input.

Output format :

The resulting array is printed on screen.

Sample test cases :

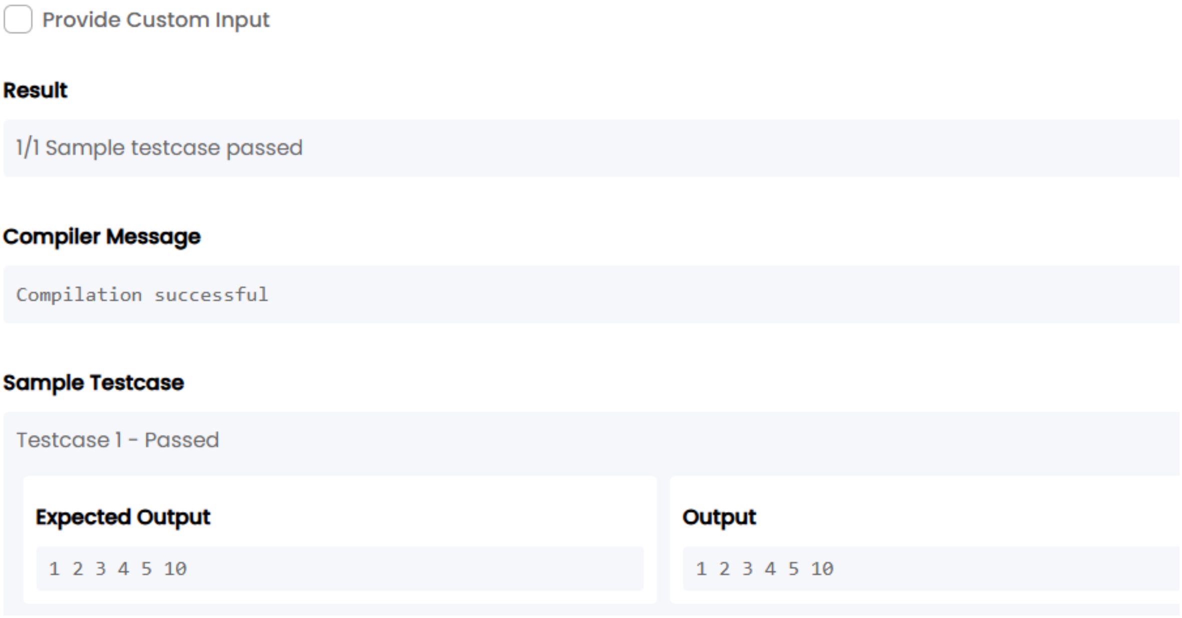
Input 1 :

```
1 2 3 4 5 10
```

Output 1 :

```
1 2 3 4 5 10
```

```
import java.util.*;
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int[] a = new int[6];
                for(int i=0;i<5;i++)
            a[i] = sc.nextInt();
                a[5] = sc.nextInt();
                for(int i=0;i<6;i++){
            System.out.print(a[i]);
            if(i<5) System.out.print(" ");
        }
    }
}
```



☐ Provide Custom Input

Result

1/1 Sample testcase passed

Compiler Message

```
Compilation successful
```

Sample Testcase

Testcase 1 - Passed

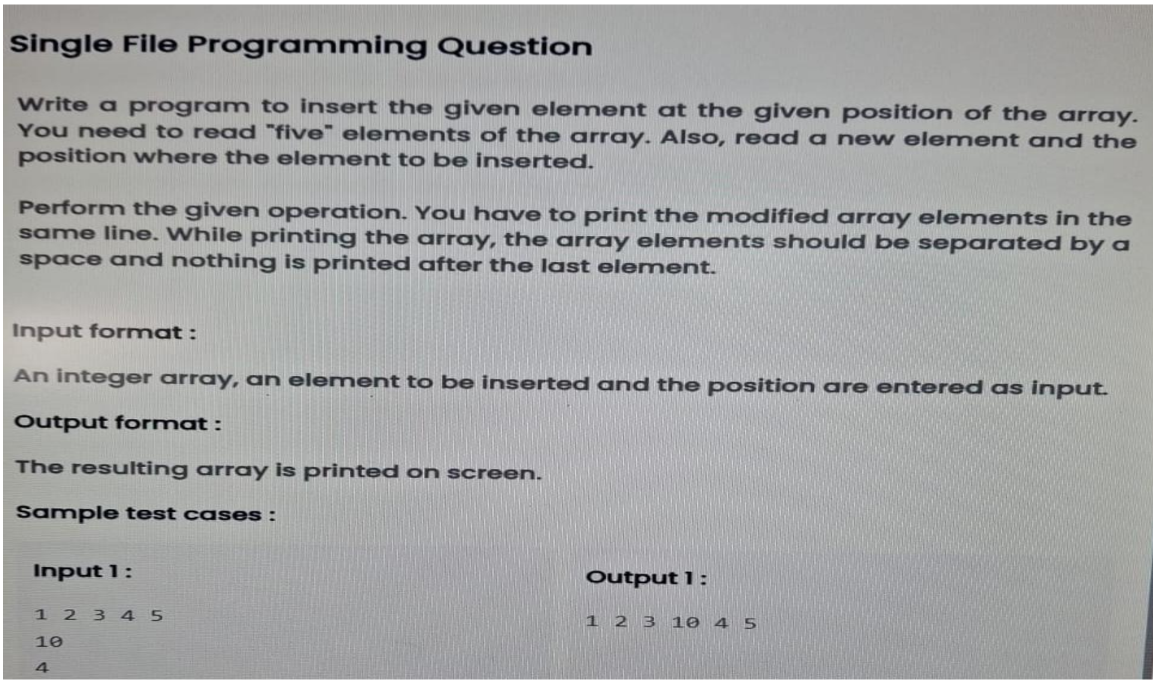
Expected Output

```
1 2 3 4 5 10
```

Output

```
1 2 3 4 5 10
```

TASK4:



Single File Programming Question

Write a program to insert the given element at the given position of the array. You need to read "five" elements of the array. Also, read a new element and the position where the element to be inserted.

Perform the given operation. You have to print the modified array elements in the same line. While printing the array, the array elements should be separated by a space and nothing is printed after the last element.

Input format :

An integer array, an element to be inserted and the position are entered as input.

Output format :

The resulting array is printed on screen.

Sample test cases :

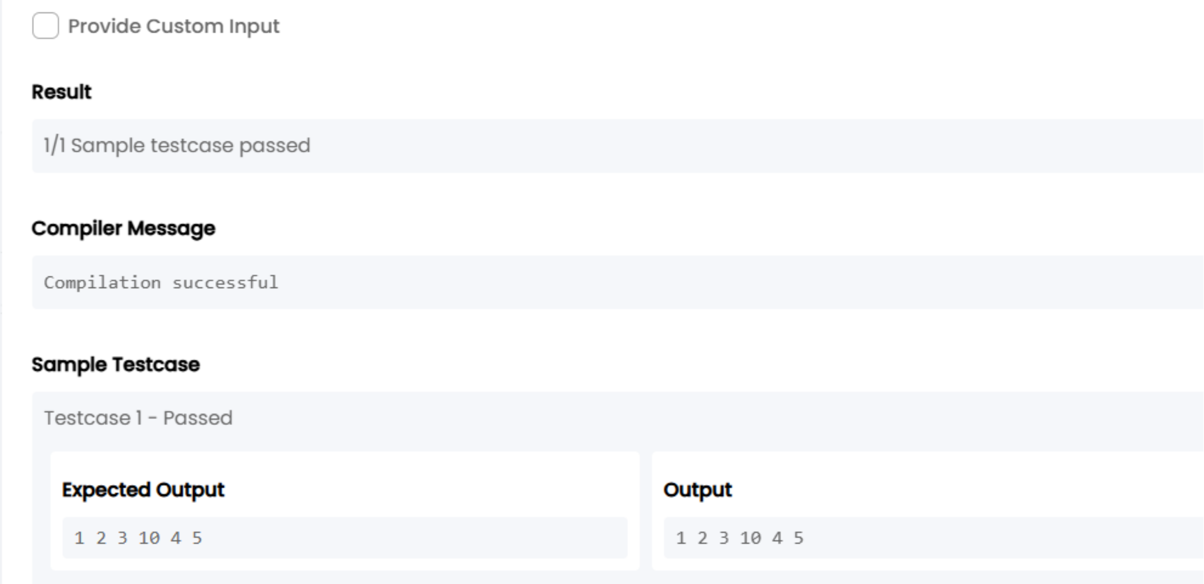
Input 1 :

```
1 2 3 4 5
10
4
```

Output 1 :

```
1 2 3 10 4 5
```

```
import java.util.*;
public class Main {
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int[] a = new int[5];
    for(int i = 0; i < 5; i++)
      a[i] = sc.nextInt();
    int x = sc.nextInt();
    int pos = sc.nextInt();
    int[] b = new int[6];
    for(int i = 0, j = 0; i < 6; i++) {
      if(i == pos - 1)
        b[i] = x;
      else
        b[i] = a[j++];
    }
    for(int i = 0; i < 6; i++) {
      System.out.print(b[i]);
      if(i < 5) System.out.print(" ")} }
}
```



☐ Provide Custom Input

Result

1/1 Sample testcase passed

Compiler Message

```
Compilation successful
```

Sample Testcase

Testcase 1 - Passed

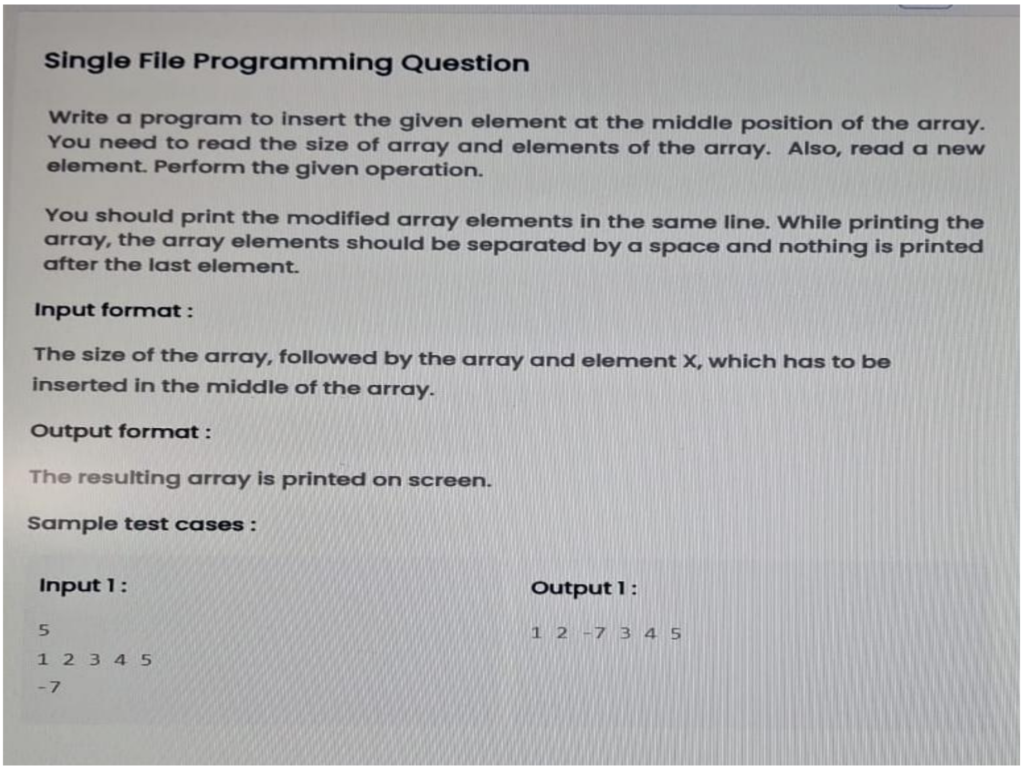
Expected Output

```
1 2 3 10 4 5
```

Output

```
1 2 3 10 4 5
```

TASK5:



Single File Programming Question

Write a program to insert the given element at the middle position of the array. You need to read the size of array and elements of the array. Also, read a new element. Perform the given operation.

You should print the modified array elements in the same line. While printing the array, the array elements should be separated by a space and nothing is printed after the last element.

Input format :

The size of the array, followed by the array and element X, which has to be inserted in the middle of the array.

Output format :

The resulting array is printed on screen.

Sample test cases :

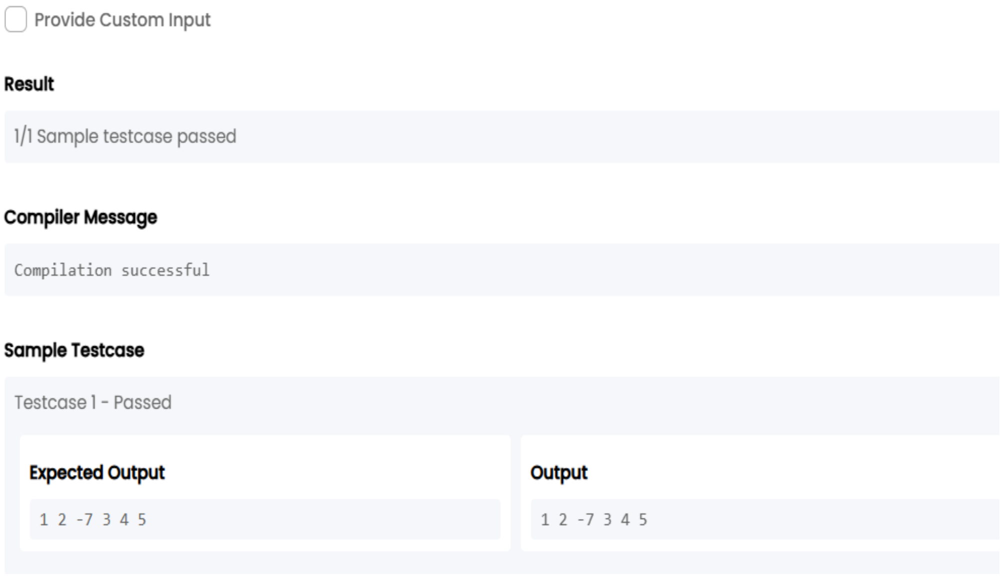
Input 1 :

```
5
1 2 3 4 5
-7
```

Output 1 :

```
1 2 -7 3 4 5
```

```
import java.util.*;
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[] a = new int[n];
            for(int i = 0; i < n; i++)
          a[i] = sc.nextInt();
            int x = sc.nextInt();
        int[] b = new int[n + 1];
        int mid = n / 2;
        for(int i = 0, j = 0; i < n + 1; i++) {
          if(i == mid)
            b[i] = x;
          else
            b[i] = a[j++];
        }
        for(int i = 0; i < n + 1; i++) {
          System.out.print(b[i]);
          if(i < n) System.out.print(" ");}}}
```



□ Provide Custom Input

Result

1/1 Sample testcase passed

Compiler Message

```
Compilation successful
```

Sample Testcase

Testcase 1 - Passed

Expected Output

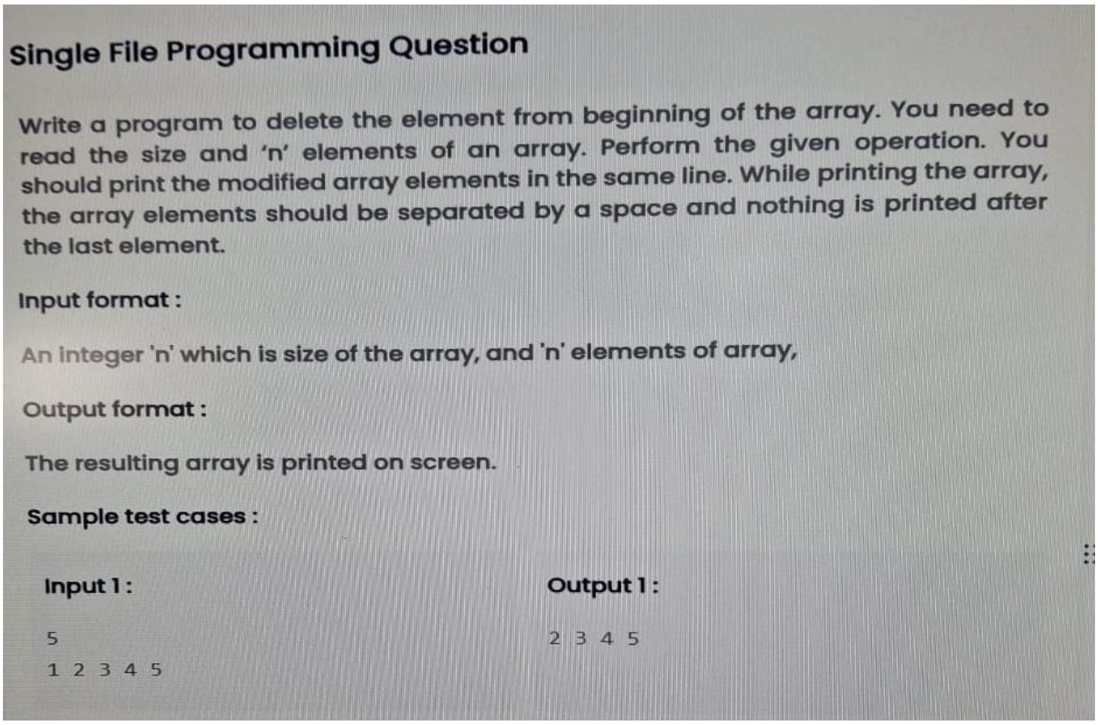
```
1 2 -7 3 4 5
```

Output

```
1 2 -7 3 4 5
```

2. Data structure 1_Arrays_Chapter 2

TASK6:



Single File Programming Question

Write a program to delete the element from beginning of the array. You need to read the size and 'n' elements of an array. Perform the given operation. You should print the modified array elements in the same line. While printing the array, the array elements should be separated by a space and nothing is printed after the last element.

Input format :

An integer 'n' which is size of the array, and 'n' elements of array,

Output format :

The resulting array is printed on screen.

Sample test cases :

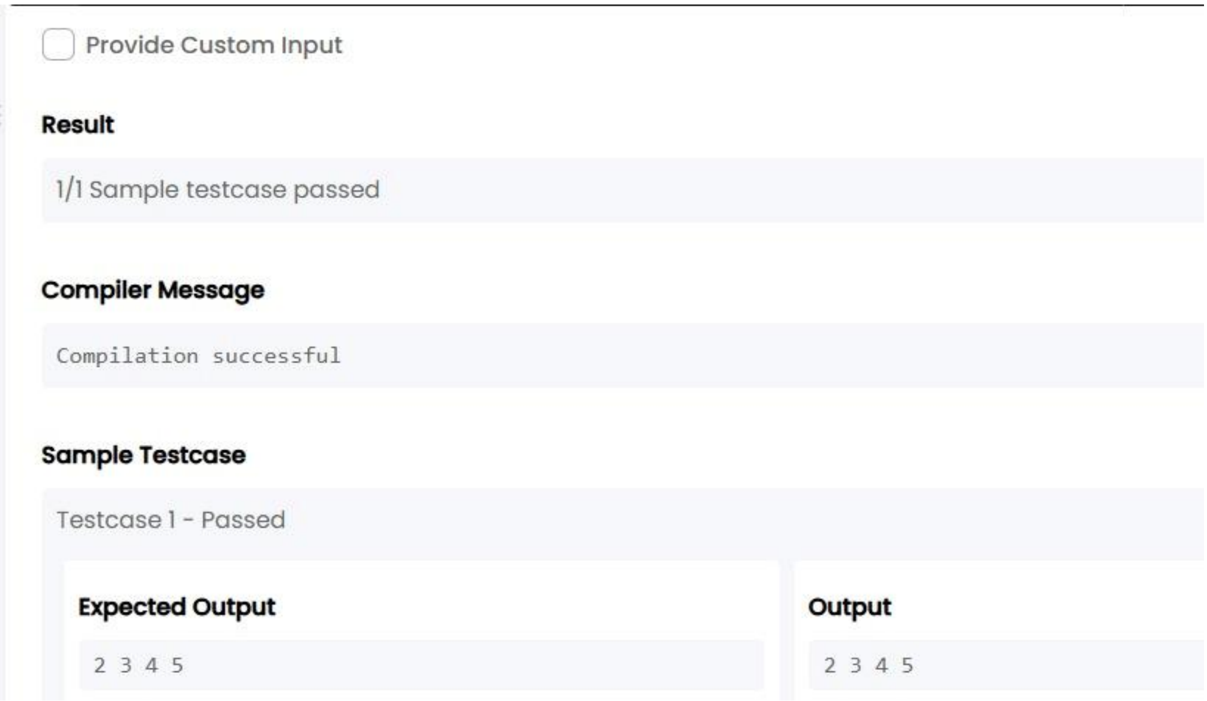
Input 1 :

```
5
1 2 3 4 5
```

Output 1 :

```
2 3 4 5
```

```
import java.util.*;
public class Main {
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
    int[] a = new int[n];
         for(int i = 0; i < n; i++)
      a[i] = sc.nextInt();
         for(int i = 1; i < n; i++) {
      System.out.print(a[i]);
      if(i < n-1) System.out.print(" ");
    }
  }
}
```



□ Provide Custom Input

Result

1/1 Sample testcase passed

Compiler Message

```
Compilation successful
```

Sample Testcase

Testcase 1 - Passed

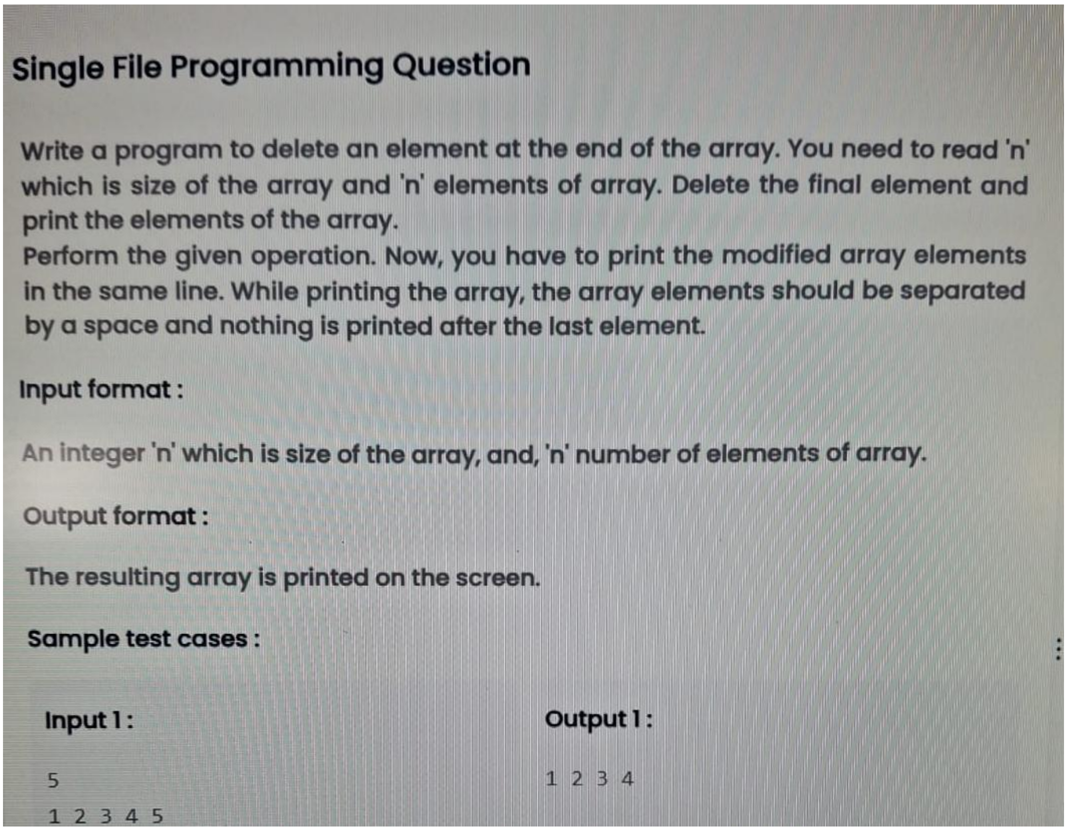
Expected Output

```
2 3 4 5
```

Output

```
2 3 4 5
```

TASK7:



Single File Programming Question

Write a program to delete an element at the end of the array. You need to read 'n' which is size of the array and 'n' elements of array. Delete the final element and print the elements of the array.

Perform the given operation. Now, you have to print the modified array elements in the same line. While printing the array, the array elements should be separated by a space and nothing is printed after the last element.

Input format :

An integer 'n' which is size of the array, and, 'n' number of elements of array.

Output format :

The resulting array is printed on the screen.

Sample test cases :

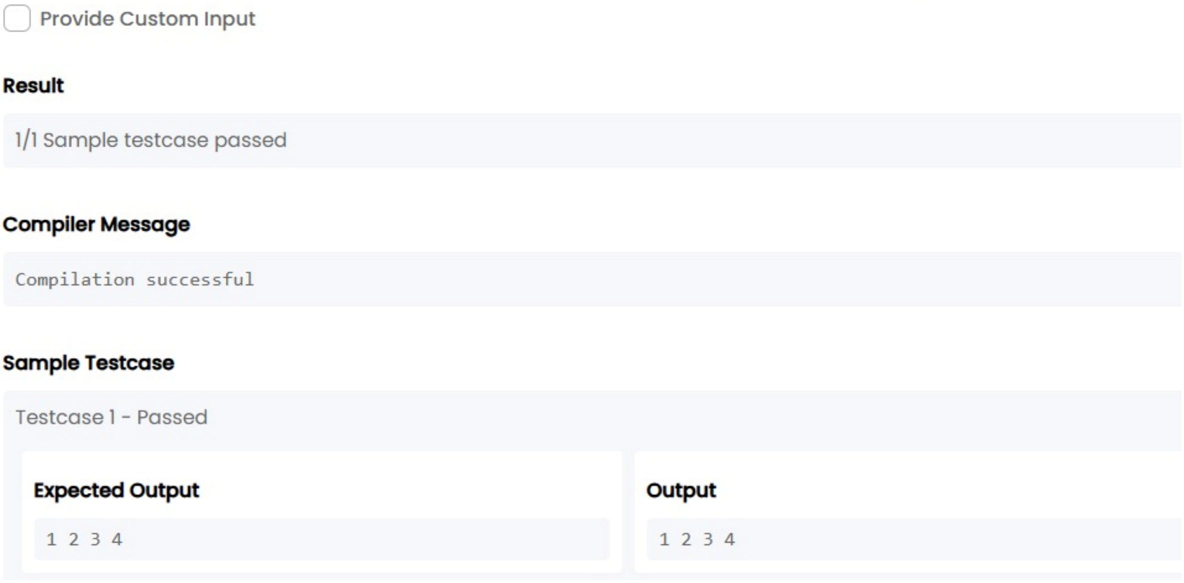
Input 1 :

```
5
1 2 3 4 5
```

Output 1 :

```
1 2 3 4
```

```
import java.util.Scanner;
public class Main {
  public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
            int n = sc.nextInt();
        int[] a = new int[n];
            for(int i = 0; i < n; i++)
          a[i] = sc.nextInt();
            for(int i = 0; i < n - 1; i++) {
          System.out.print(a[i]);
          if(i < n - 2)
            System.out.print(" ");
        }
      }
    }
```



□ Provide Custom Input

Result

1/1 Sample testcase passed

Compiler Message

```
Compilation successful
```

Sample Testcase

Testcase 1 - Passed

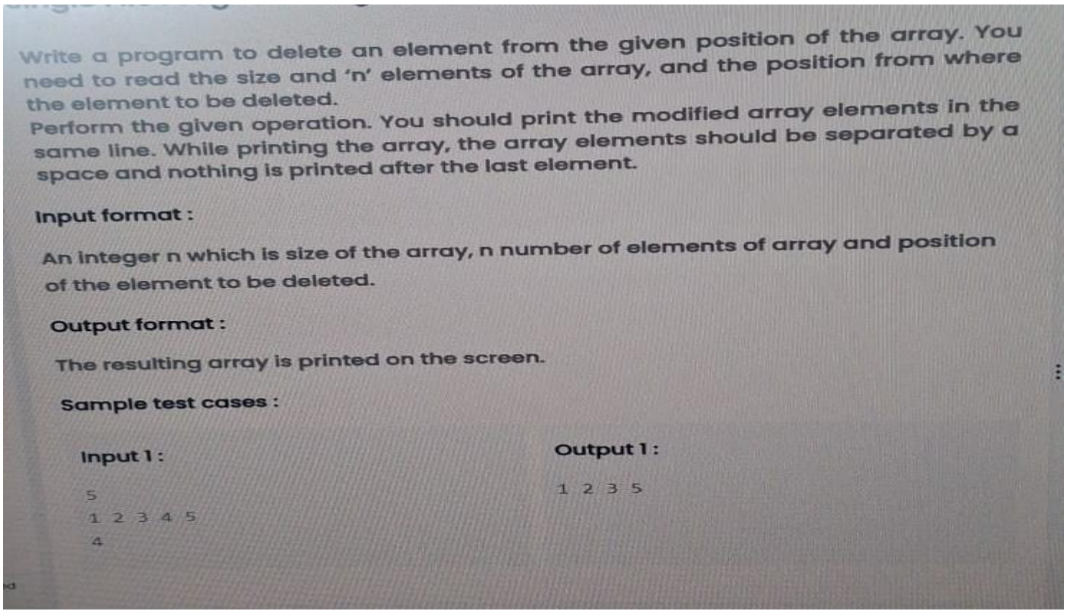
Expected Output

```
1 2 3 4
```

Output

```
1 2 3 4
```

TASK8:



Write a program to delete an element from the given position of the array. You need to read the size and 'n' elements of the array, and the position from where the element to be deleted.
Perform the given operation. You should print the modified array elements in the same line. While printing the array, the array elements should be separated by a space and nothing is printed after the last element.

Input format :

An integer n which is size of the array, n number of elements of array and position of the element to be deleted.

Output format :

The resulting array is printed on the screen.

Sample test cases :

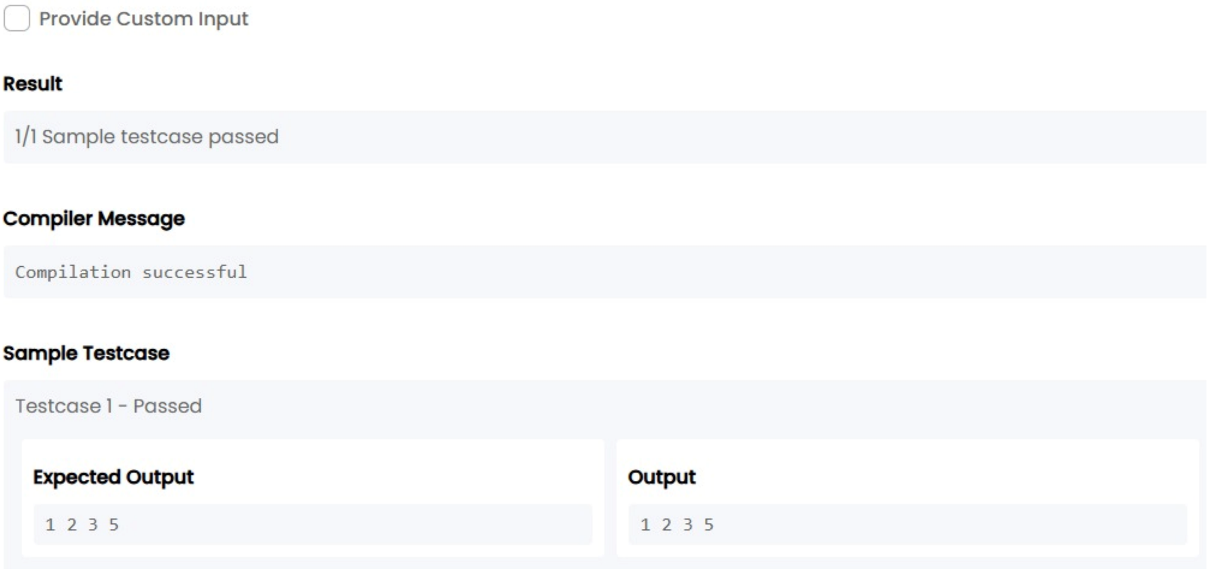
Input 1 :

```
5
1 2 3 4 5
4
```

Output 1 :

```
1 2 3 5
```

```java
import java.util.Scanner;

public class Main {

   public static void main(String[] args) {

      Scanner sc = new Scanner(System.in);

      int n = sc.nextInt();

      int[] a = new int[n];

      for(int i = 0; i < n; i++)

         a[i] = sc.nextInt();

      int pos = sc.nextInt(); // 1-based position

      for(int i = 0; i < n; i++) {

         if(i != pos - 1) {

            System.out.print(a[i]);

            if(i < n - 1 && !(i == n - 2 && pos == n))

               System.out.print(" ");

         }

      }

   }

}
```



□ Provide Custom Input

Result

1/1 Sample testcase passed

Compiler Message

```
Compilation successful
```

Sample Testcase

Testcase 1 - Passed

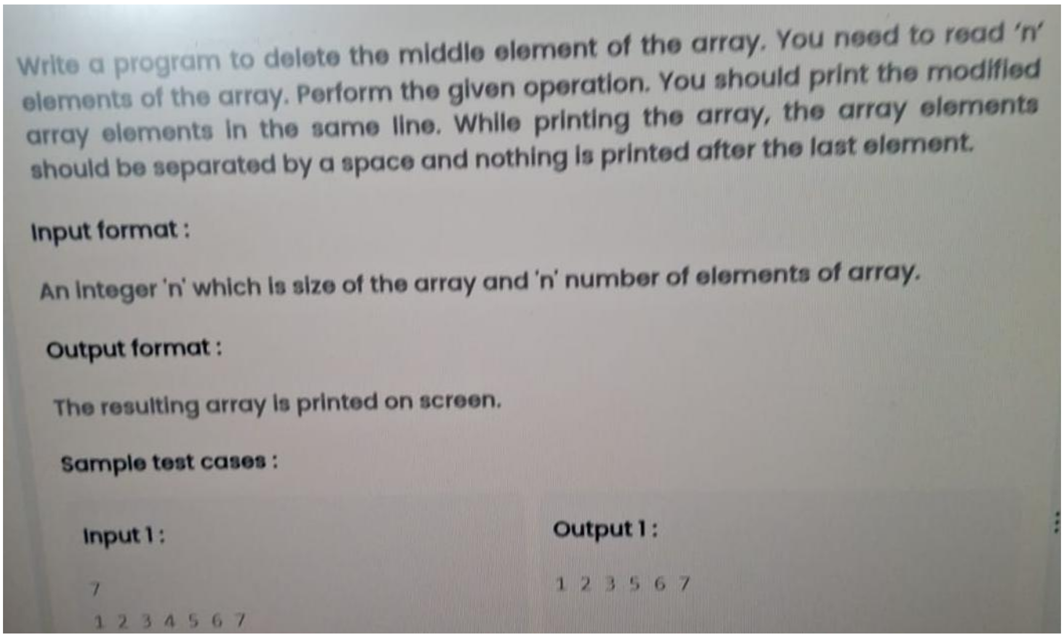
Expected Output

```
1 2 3 5
```

Output

```
1 2 3 5
```

TASK9:



Write a program to delete the middle element of the array. You need to read 'n' elements of the array. Perform the given operation. You should print the modified array elements in the same line. While printing the array, the array elements should be separated by a space and nothing is printed after the last element.

Input format :

An integer 'n' which is size of the array and 'n' number of elements of array.

Output format :

The resulting array is printed on screen.

Sample test cases :

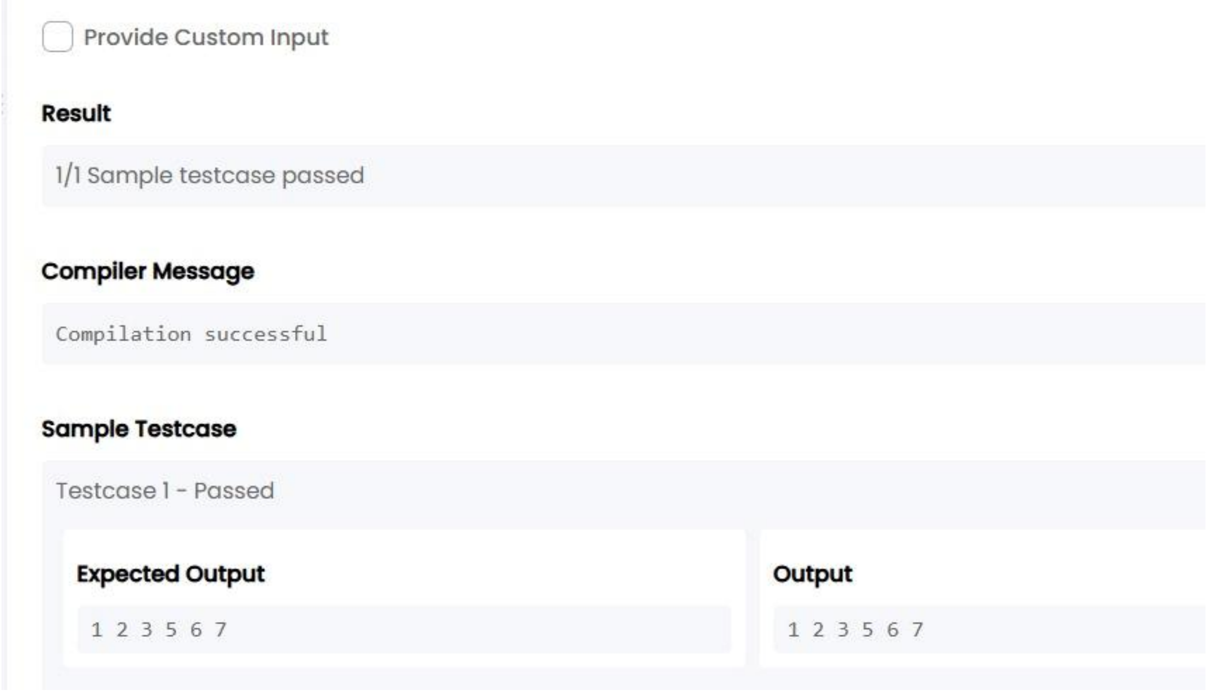
Input 1 :

```
7
1 2 3 4 5 6 7
```

Output 1 :

```
1 2 3 5 6 7
```

```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
              int n = sc.nextInt();
        int[] a = new int[n];
               for(int i = 0; i < n; i++)
            a[i] = sc.nextInt();
               int mid = n / 2;  // middle index
        for(int i = 0; i < n; i++) {
      if(i != mid) {
        System.out.print(a[i]);
        if(i != n - 1 && !(i == n - 2 && mid == n - 1))
          System.out.print(" ");
      }
    }
  }
}
```



□ Provide Custom Input

Result

1/1 Sample testcase passed

Compiler Message

```
Compilation successful
```

Sample Testcase

Testcase 1 - Passed

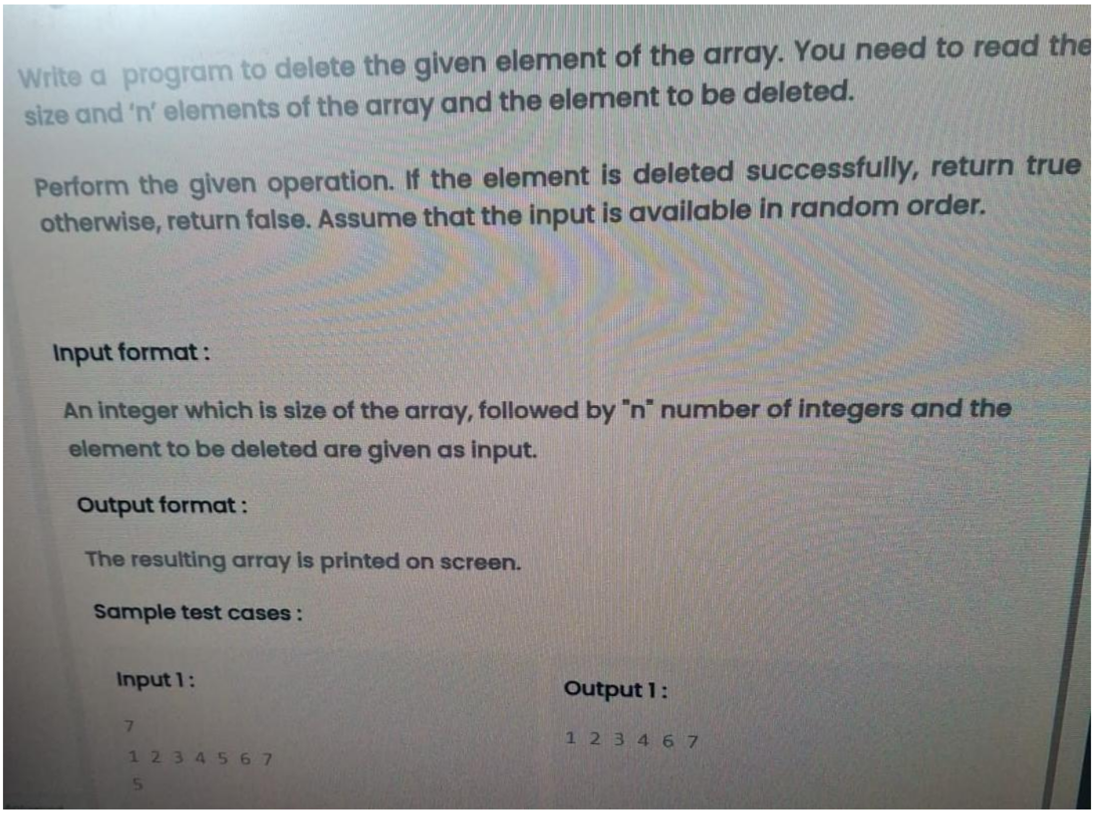
Expected Output

```
1 2 3 5 6 7
```

Output

```
1 2 3 5 6 7
```

TASK10:



Write a program to delete the given element of the array. You need to read the size and 'n' elements of the array and the element to be deleted.

Perform the given operation. If the element is deleted successfully, return true otherwise, return false. Assume that the input is available in random order.

Input format :

An integer which is size of the array, followed by "n" number of integers and the element to be deleted are given as input.

Output format :

The resulting array is printed on screen.

Sample test cases :

Input 1 :

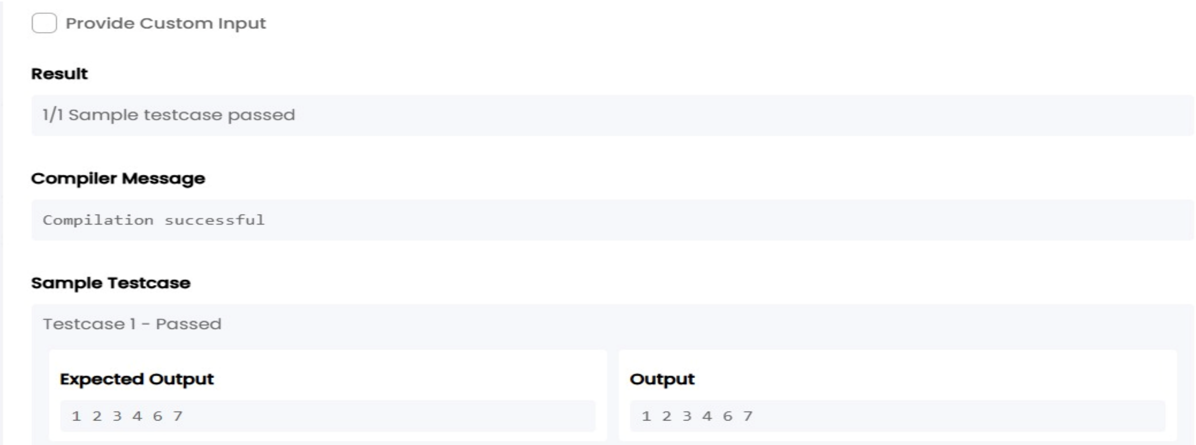
```
7
1 2 3 4 5 6 7
5
```

Output 1 :

```
1 2 3 4 6 7
```

```
import java.util.Scanner;
public class Main {
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int n = sc.nextInt();
    int[] a = new int[n];
    for(int i = 0; i < n; i++)
      a[i] = sc.nextInt();
    int del = sc.nextInt();
    boolean found = false;
    for(int i = 0; i < n; i++){
      if(a[i] == del && !found){
        found = true;
        continue;
      }
      System.out.print(a[i]);
```

```
            if(i < n - 1)
                System.out.print(" ");
        }
        if(!found)
            System.out.print("false");
    }
}
```



Result

1/1 Sample testcase passed

Compiler Message

```
Compilation successful
```

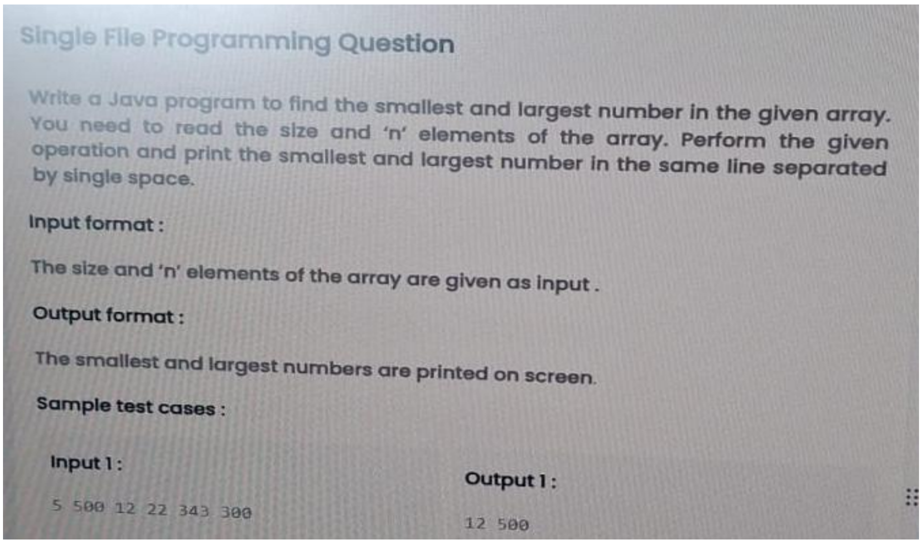
Sample Testcase

Testcase 1 - Passed

Expected Output	Output
1 2 3 4 6 7	1 2 3 4 6 7

3. Data structure 4_Search & Sorting_Chapter1

TASK11:



Single File Programming Question

Write a Java program to find the smallest and largest number in the given array. You need to read the size and 'n' elements of the array. Perform the given operation and print the smallest and largest number in the same line separated by single space.

Input format :

The size and 'n' elements of the array are given as input .

Output format :

The smallest and largest numbers are printed on screen.

Sample test cases :

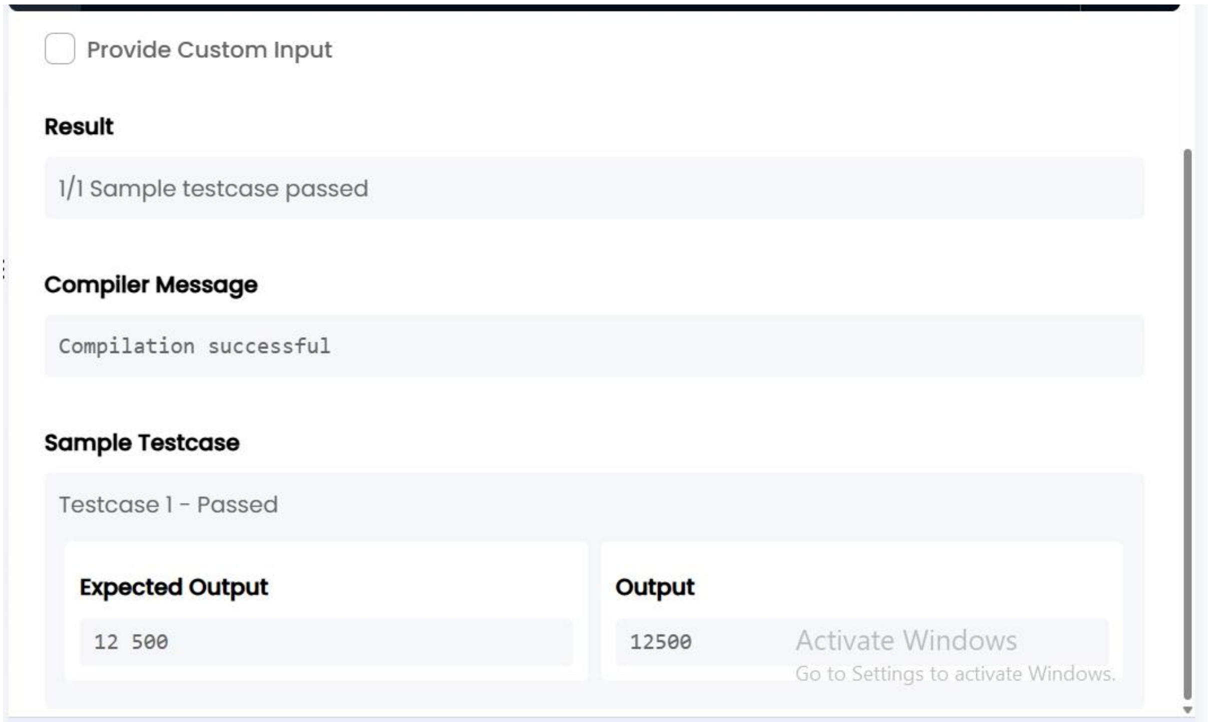
Input 1 :

```
5 500 12 22 343 300
```

Output 1 :

```
12 500
```

```
import java.util.Scanner;

public class Main {

   public static void main(String[] args) {

      Scanner sc = new Scanner(System.in);

      int n = sc.nextInt();

      int min = Integer.MAX_VALUE;

      int max = Integer.MIN_VALUE;

      for(int i = 0; i < n; i++){

         int num = sc.nextInt();

         if(num < min) min = num;

         if(num > max) max = num;

      }

      System.out.print(min + " " + max);

   }

}
```



□ Provide Custom Input

Result

1/1 Sample testcase passed

Compiler Message

```
Compilation successful
```

Sample Testcase

Testcase 1 - Passed

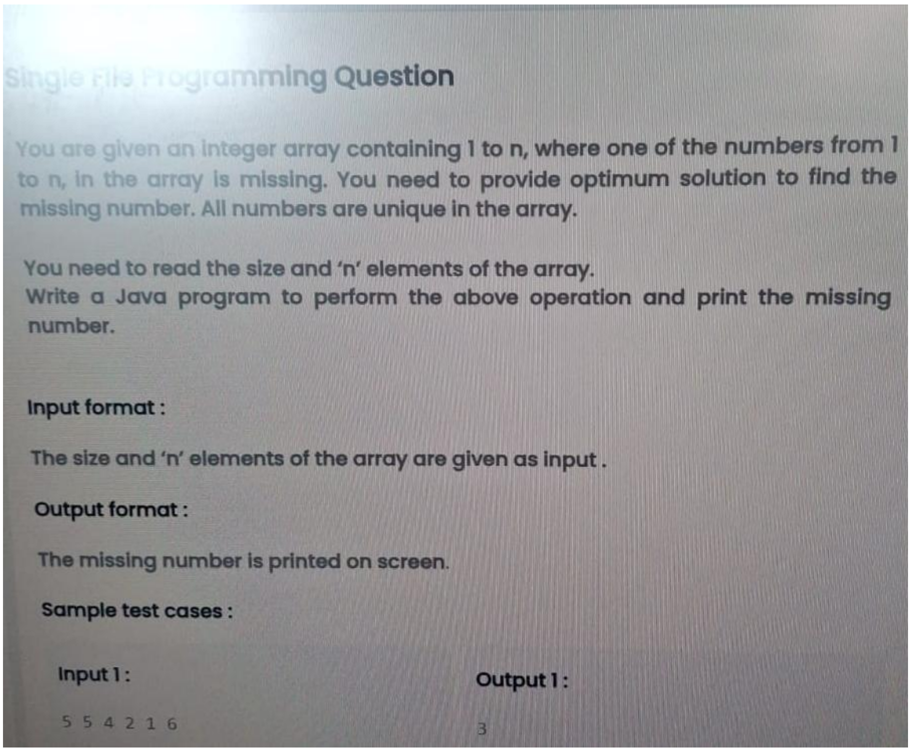
Expected Output

```
12 500
```

Output

```
12500
```

TASK12:



Single File Programming Question

You are given an integer array containing 1 to n, where one of the numbers from 1 to n, in the array is missing. You need to provide optimum solution to find the missing number. All numbers are unique in the array.

You need to read the size and 'n' elements of the array.
Write a Java program to perform the above operation and print the missing number.

Input format :

The size and 'n' elements of the array are given as input .

Output format :

The missing number is printed on screen.

Sample test cases :

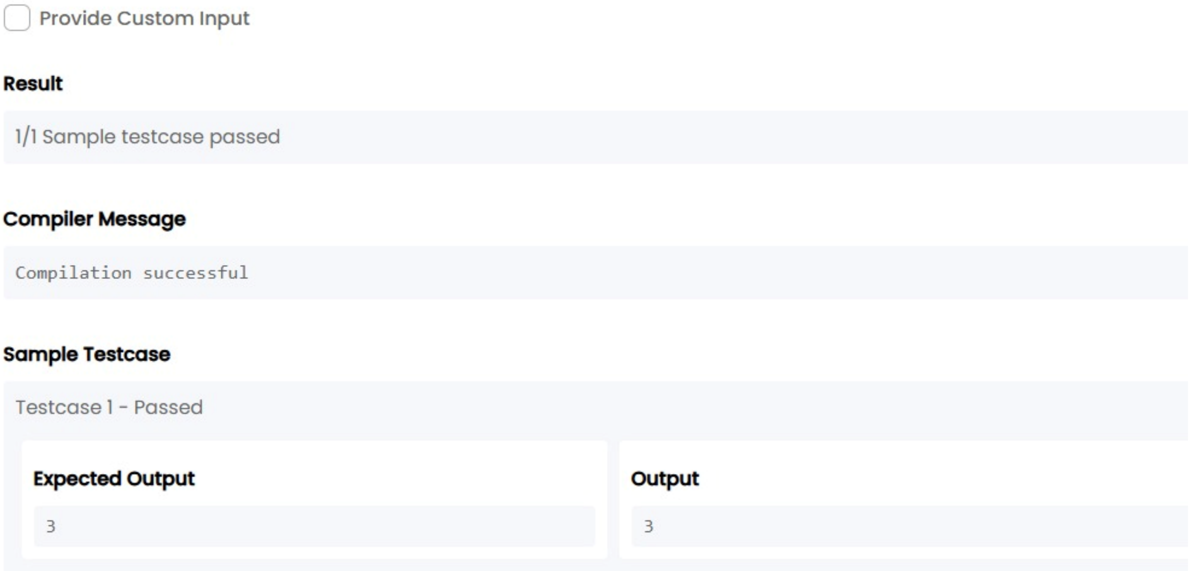
Input 1 :

```
5 5 4 2 1 6
```

Output 1 :

```
3
```

```
import java.util.Scanner;
import java.util.Arrays;
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[] a = new int[n];
        for(int i = 0; i < n; i++)
            a[i] = sc.nextInt();
        Arrays.sort(a);
        for(int i = 0; i < n - 1; i++) {
            if(a[i+1] - a[i] != 1) {
                System.out.print(a[i] + 1);
                return;
            }
        }
    }
}
```



☐ Provide Custom Input

Result

1/1 Sample testcase passed

Compiler Message

```
Compilation successful
```

Sample Testcase

Testcase 1 - Passed

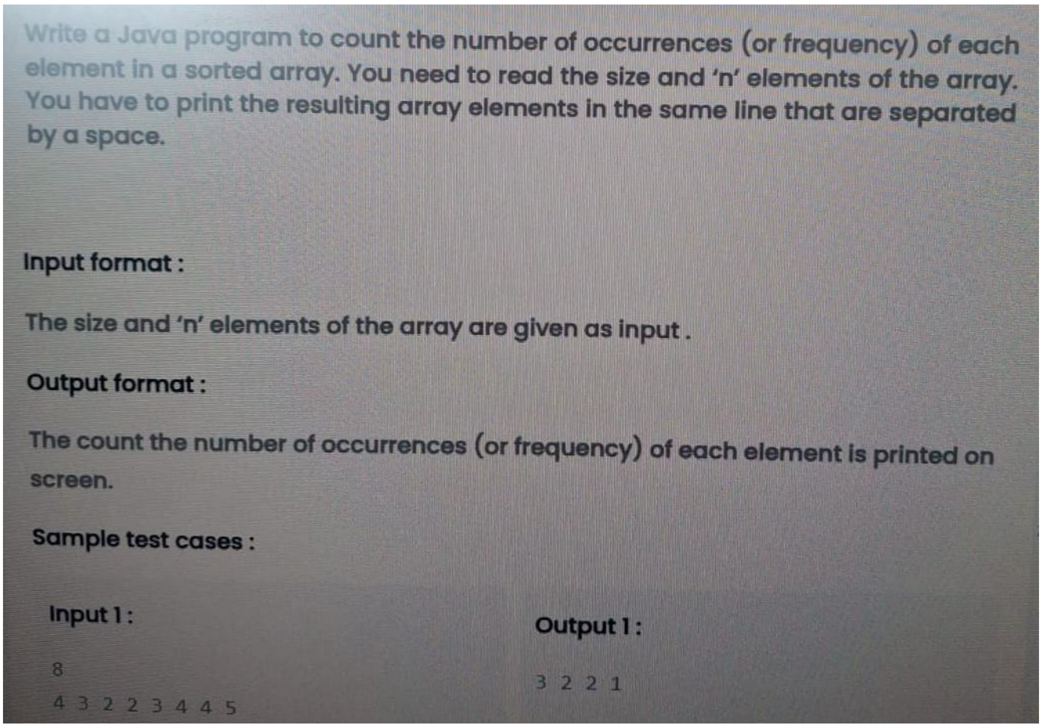
Expected Output

```
3
```

Output

```
3
```

TASK13:



Write a Java program to count the number of occurrences (or frequency) of each element in a sorted array. You need to read the size and 'n' elements of the array. You have to print the resulting array elements in the same line that are separated by a space.

Input format :

The size and 'n' elements of the array are given as input .

Output format :

The count the number of occurrences (or frequency) of each element is printed on screen.

Sample test cases :

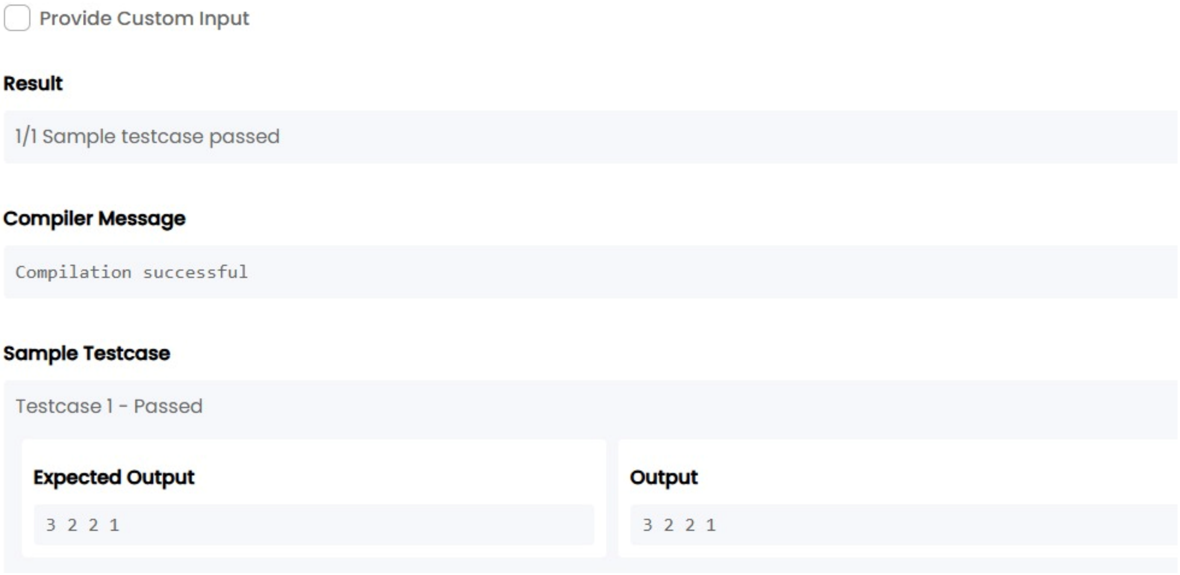
Input 1 :

```
8
4 3 2 2 3 4 4 5
```

Output 1 :

```
3 2 2 1
```

```
import java.util.*;
public class Main {
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
    int[] a = new int[n];
        for(int i = 0; i < n; i++)
      a[i] = sc.nextInt();
        LinkedHashMap<Integer,Integer> map = new LinkedHashMap<>();
        for(int i = 0; i < n; i++)
      map.put(a[i], map.getOrDefault(a[i],0)+1);
        for(int val : map.values())
      System.out.print(val + " ");
  }
}
```



□ Provide Custom Input

Result

1/1 Sample testcase passed

Compiler Message

```
Compilation successful
```

Sample Testcase

Testcase 1 - Passed

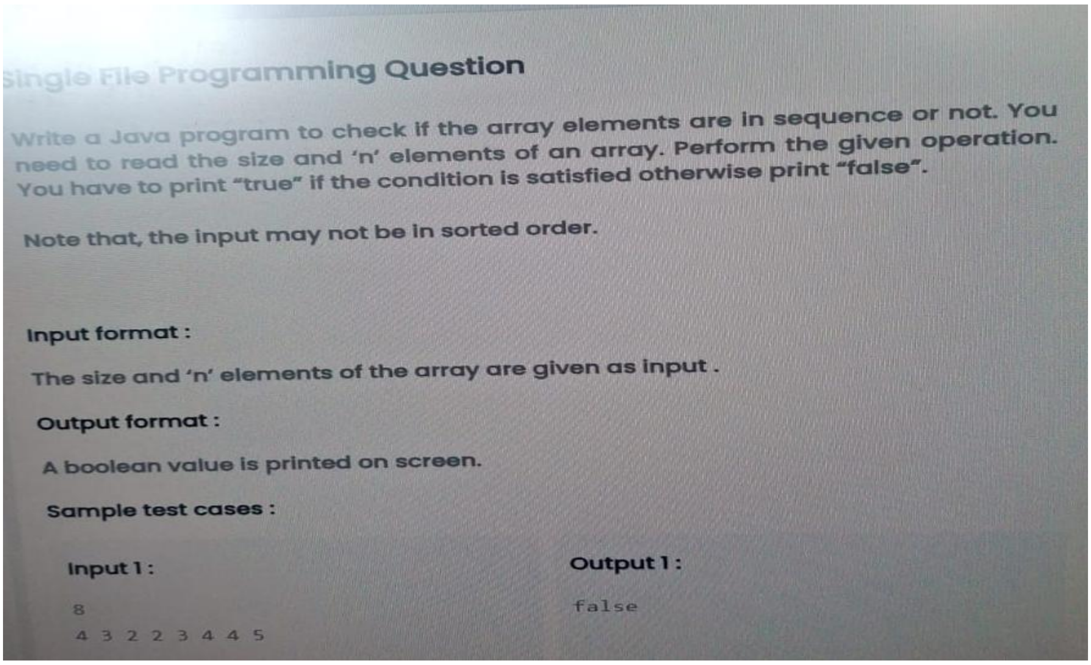
Expected Output

```
3 2 2 1
```

Output

```
3 2 2 1
```

TASK14:



Single File Programming Question

Write a Java program to check if the array elements are in sequence or not. You need to read the size and 'n' elements of an array. Perform the given operation. You have to print "true" if the condition is satisfied otherwise print "false".

Note that, the input may not be in sorted order.

Input format :

The size and 'n' elements of the array are given as input .

Output format :

A boolean value is printed on screen.

Sample test cases :

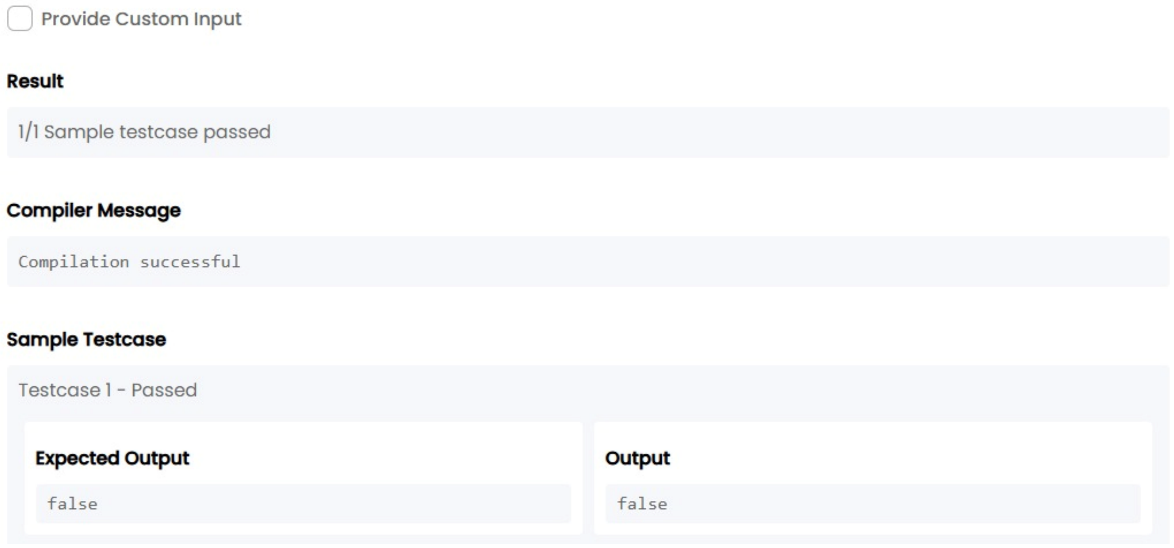
Input 1 :

```
8
4 3 2 2 3 4 4 5
```

Output 1 :

```
false
```

```
import java.util.*;

public class Main {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int n = sc.nextInt();

        int[] a = new int[n];

        for(int i = 0; i < n; i++)

            a[i] = sc.nextInt();

        Arrays.sort(a);
        boolean flag = true;
        for(int i = 1; i < n; i++) {
            if(a[i] != a[i-1] + 1) {
                flag = false;
                break;
            }
        }
        System.out.print(flag);
    }
}
```



☐ Provide Custom Input

Result

1/1 Sample testcase passed

Compiler Message

```
Compilation successful
```

Sample Testcase

Testcase 1 - Passed

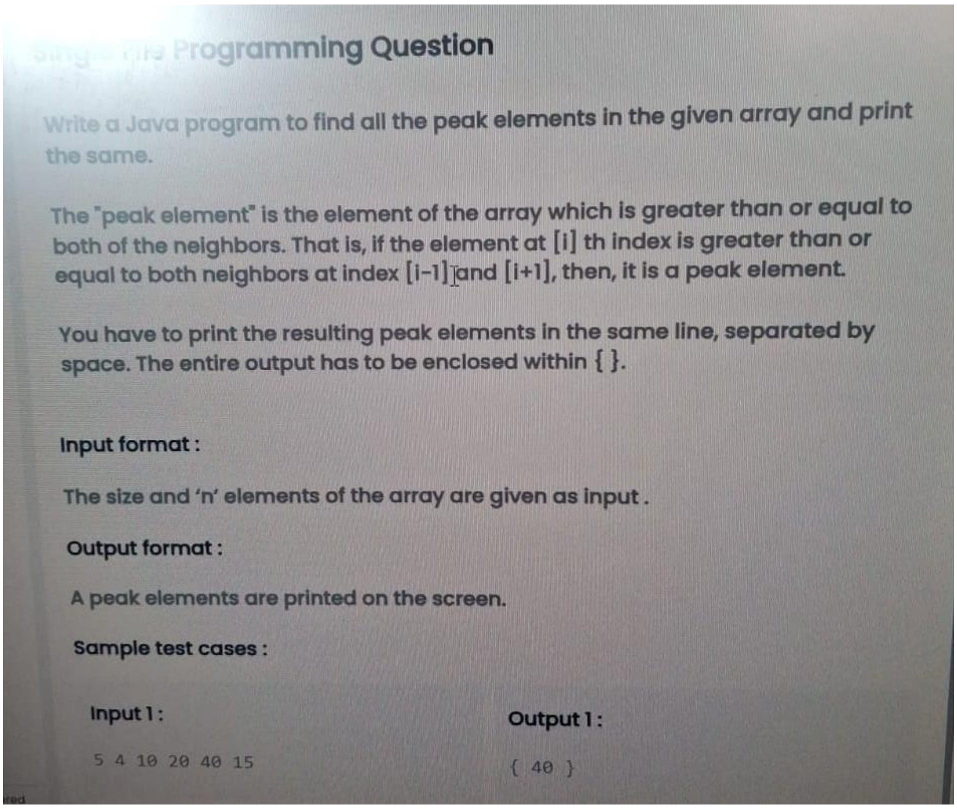
Expected Output

```
false
```

Output

```
false
```

TASK15:



Single File Programming Question

Write a Java program to find all the peak elements in the given array and print the same.

The "peak element" is the element of the array which is greater than or equal to both of the neighbors. That is, if the element at [i] th index is greater than or equal to both neighbors at index [i-1] and [i+1], then, it is a peak element.

You have to print the resulting peak elements in the same line, separated by space. The entire output has to be enclosed within { }.

Input format :

The size and 'n' elements of the array are given as input .

Output format :

A peak elements are printed on the screen.

Sample test cases :

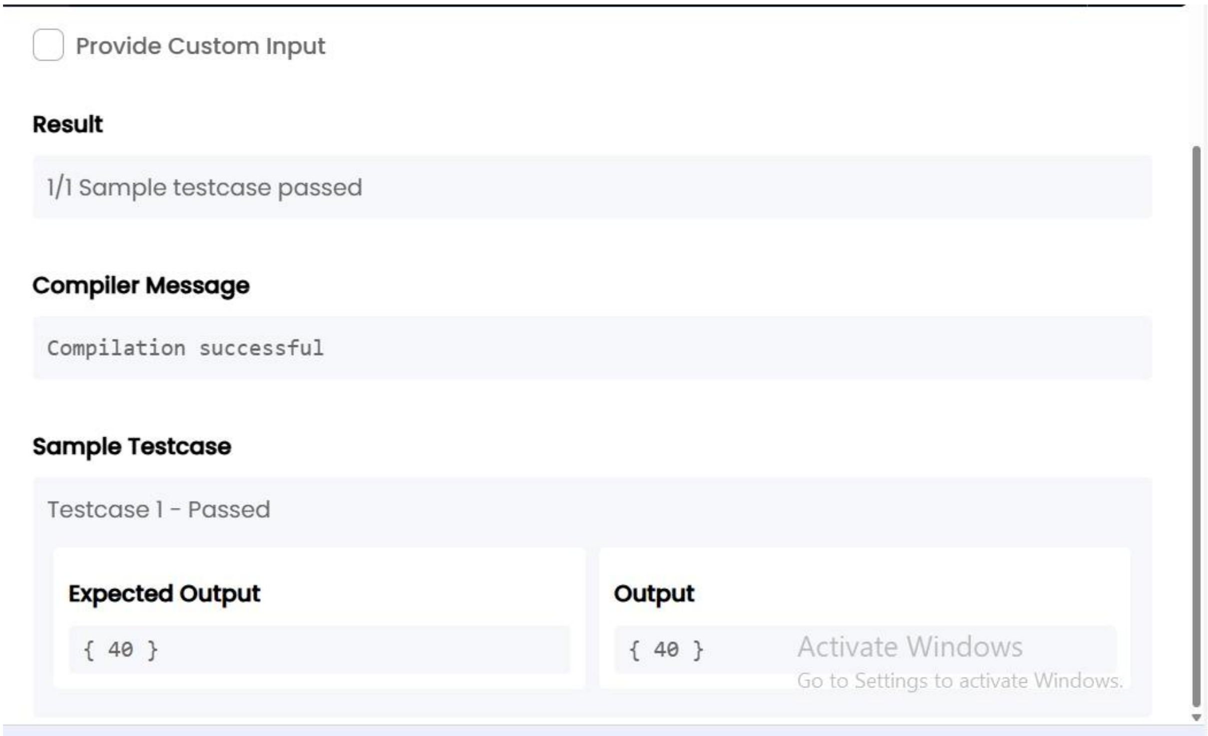
Input 1 :

```
5 4 10 20 40 15
```

Output 1 :

```
{ 40 }
```

```
import java.util.*;
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[] a = new int[n];
        for (int i = 0; i < n; i++)
            a[i] = sc.nextInt();
        System.out.print("{ ");
        boolean first = true;
        for (int i = 1; i < n - 1; i++) {
            if (a[i] >= a[i - 1] && a[i] >= a[i + 1]) {
                if (!first)
                    System.out.print(" ");
                System.out.print(a[i]);
                first = false;
        }
      }
      System.out.print(" }");
    }
}
```



□ Provide Custom Input

Result

1/1 Sample testcase passed

Compiler Message

```
Compilation successful
```

Sample Testcase

Testcase 1 - Passed

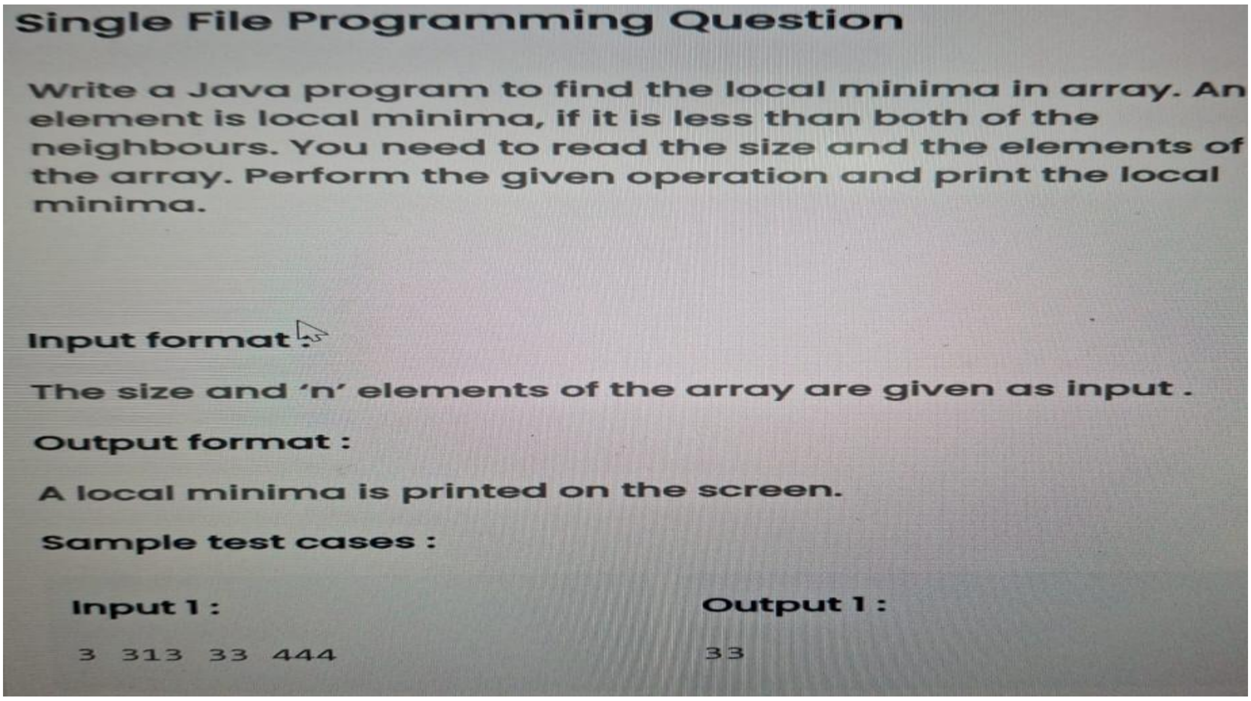
Expected Output

```
{ 40 }
```

Output

```
{ 40 }
```

TASK16:



Single File Programming Question

Write a Java program to find the local minima in array. An element is local minima, if it is less than both of the neighbours. You need to read the size and the elements of the array. Perform the given operation and print the local minima.

Input format :

The size and 'n' elements of the array are given as input .

Output format :

A local minima is printed on the screen.

Sample test cases :

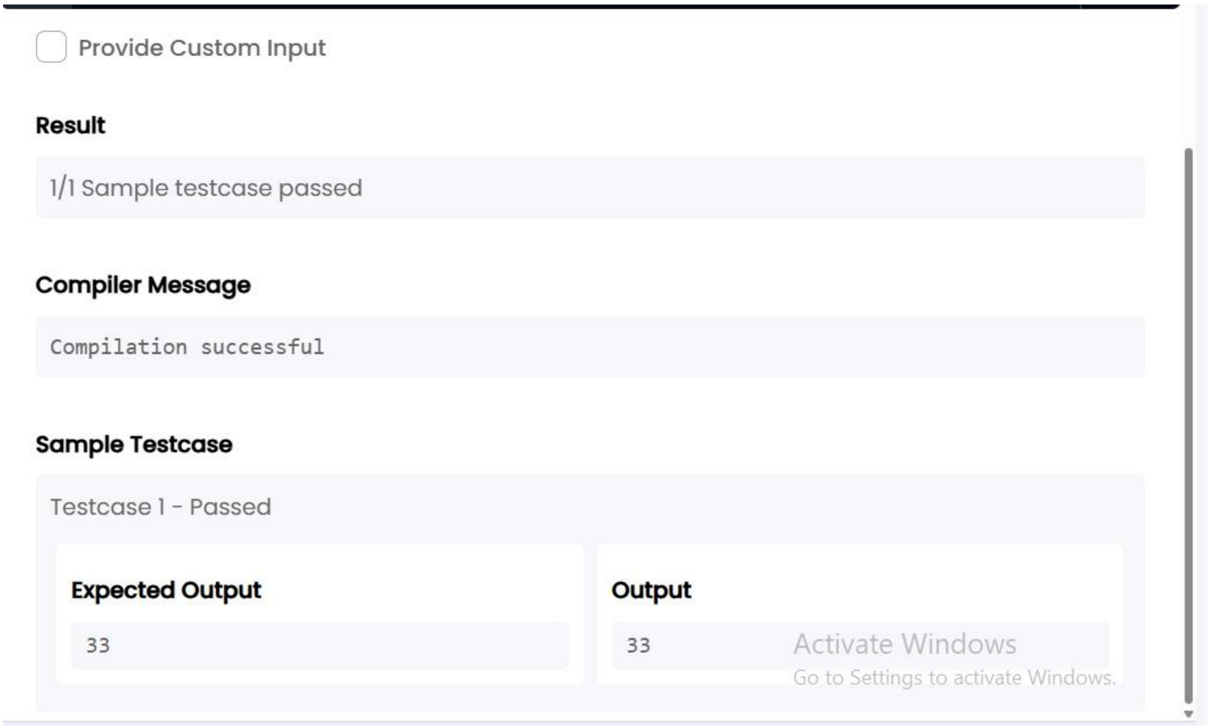
Input 1 :

```
3 313 33 444
```

Output 1 :

```
33
```

```java
import java.util.*;
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[] a = new int[n];
        for(int i=0;i<n;i++)
            a[i]=sc.nextInt();
        boolean first = true;
        if(n==1){
            System.out.print(a[0]);
            return;
        }
        // First element
        if(a[0] < a[1]){
            System.out.print(a[0]);
            first = false;
        }
        // Middle elements
        for(int i=1;i<n-1;i++){
            if(a[i] < a[i-1] && a[i] < a[i+1]){
                if(!first) System.out.print(" ");
                System.out.print(a[i]);
                first = false;
            }
        }
        // Last element
        if(a[n-1] < a[n-2]){
            if(!first) System.out.print(" ");
            System.out.print(a[n-1]);}}}
```



□ Provide Custom Input

Result

1/1 Sample testcase passed

Compiler Message

```
Compilation successful
```

Sample Testcase

Testcase 1 - Passed

Expected Output

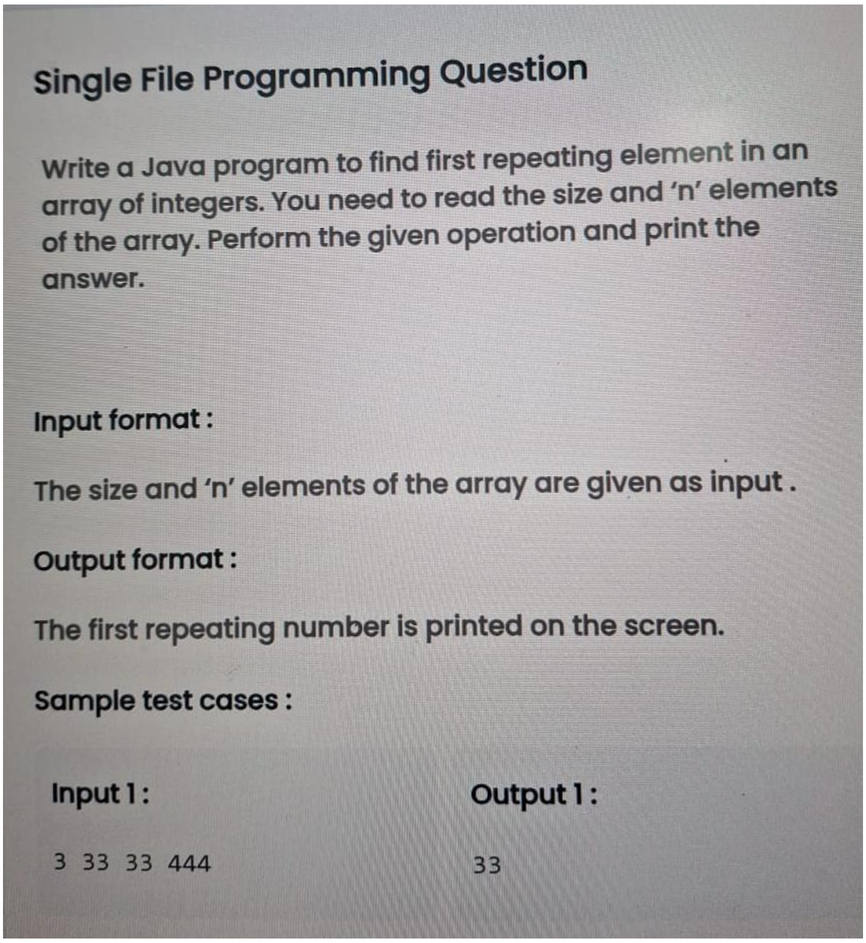
```
33
```

Output

```
33
```

4. Data structure 4_Search & Sorting_Chapter 2

TASK17:



Single File Programming Question

Write a Java program to find first repeating element in an array of integers. You need to read the size and 'n' elements of the array. Perform the given operation and print the answer.

Input format :

The size and 'n' elements of the array are given as input .

Output format :

The first repeating number is printed on the screen.

Sample test cases :

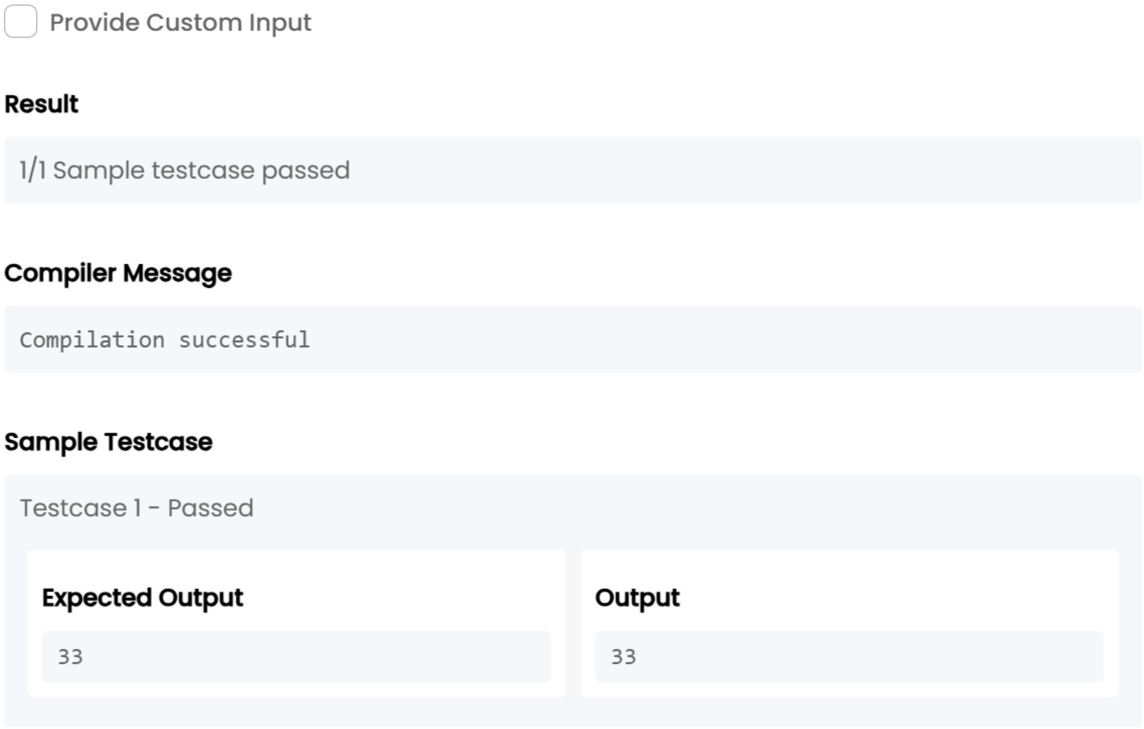
Input 1 :

```
3 33 33 444
```

Output 1 :

```
33
```

```
import java.util.*;
public class Main {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      int n = sc.nextInt();
      int[] a = new int[n];
     for(int i=0;i<n;i++)
         a[i]=sc.nextInt();
      for(int i=0;i<n;i++){
         for(int j=i+1;j<n;j++){
            if(a[i]==a[j]){
                    System.out.print(a[i]);
                    return;
                }
            }
        }
    }
}
```



☐ Provide Custom Input

Result

1/1 Sample testcase passed

Compiler Message

```
Compilation successful
```

Sample Testcase

Testcase 1 - Passed

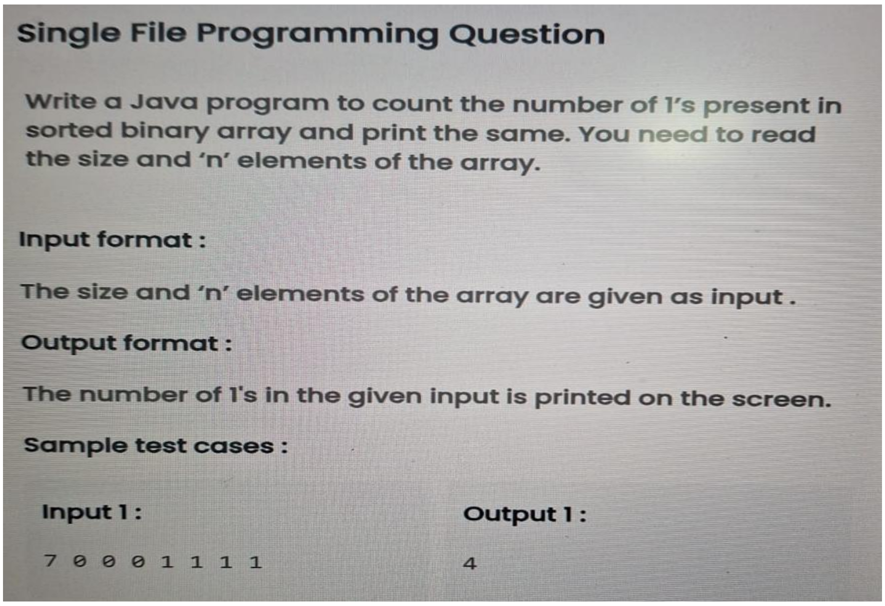
Expected Output

```
33
```

Output

```
33
```

TASK18:



Single File Programming Question

Write a Java program to count the number of 1's present in sorted binary array and print the same. You need to read the size and 'n' elements of the array.

Input format :

The size and 'n' elements of the array are given as input .

Output format :

The number of 1's in the given input is printed on the screen.

Sample test cases :

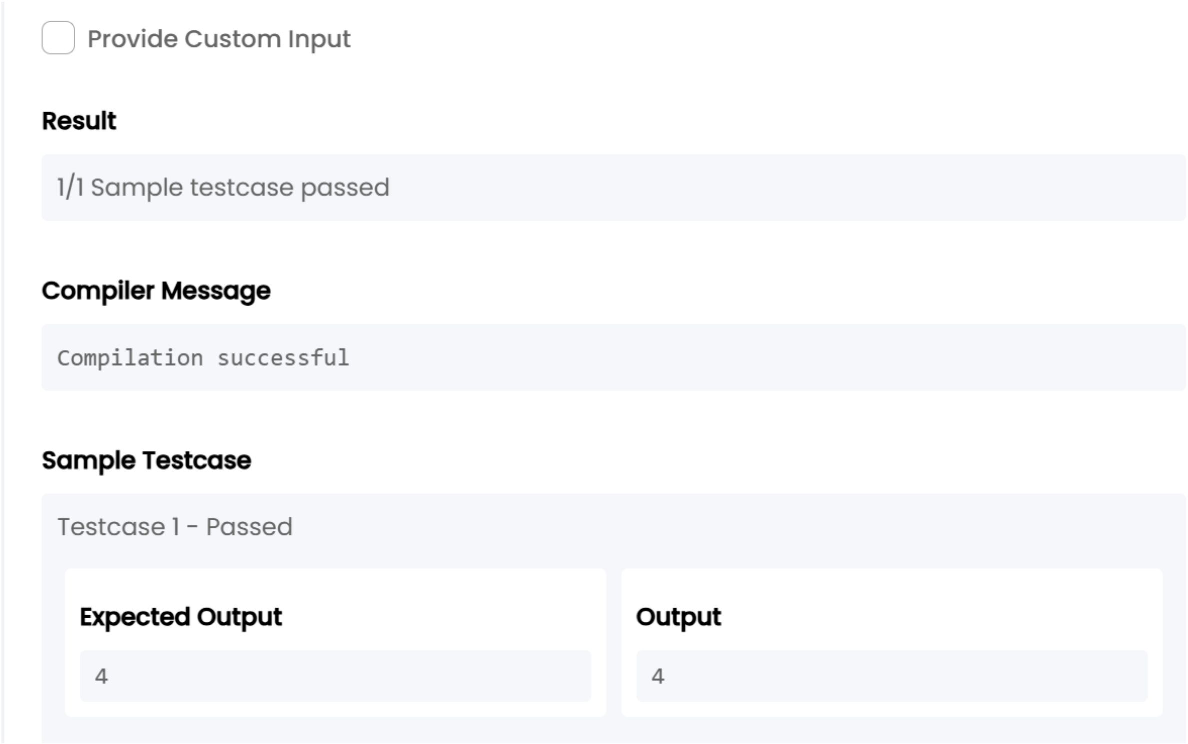
Input 1 :

```
7 0 0 0 1 1 1 1
```

Output 1 :

```
4
```

```
import java.util.*;

public class Main {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int n = sc.nextInt();

        int count = 0;

        for(int i = 0; i < n; i++)

            if(sc.nextInt() == 1)

                count++;

        System.out.print(count);

    }

}
```



□ Provide Custom Input

Result

1/1 Sample testcase passed

Compiler Message

```
Compilation successful
```

Sample Testcase

Testcase 1 - Passed

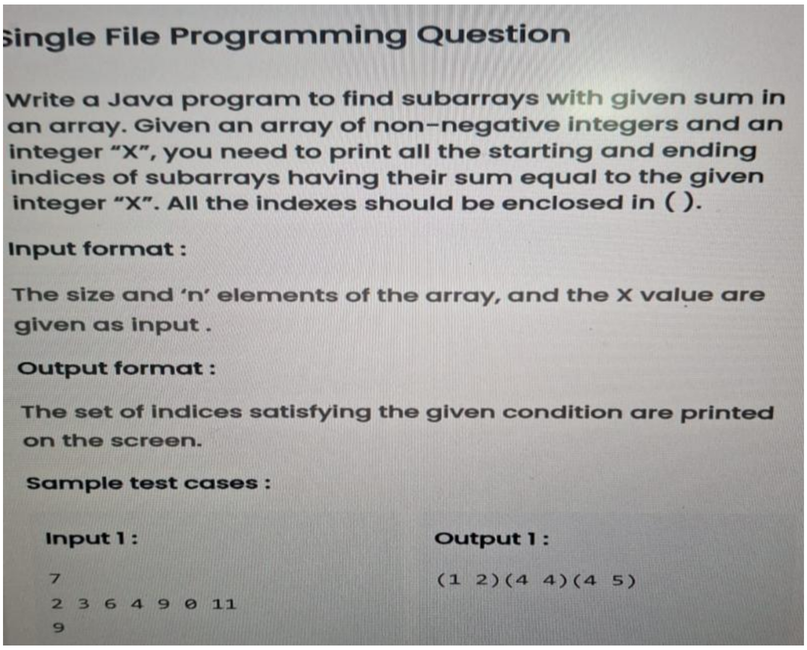
Expected Output

```
4
```

Output

```
4
```

TASK19:



Single File Programming Question

Write a Java program to find subarrays with given sum in an array. Given an array of non-negative integers and an integer "X", you need to print all the starting and ending indices of subarrays having their sum equal to the given integer "X". All the indexes should be enclosed in ( ).

Input format :

The size and 'n' elements of the array, and the X value are given as input .

Output format :

The set of indices satisfying the given condition are printed on the screen.

Sample test cases :

Input 1 :

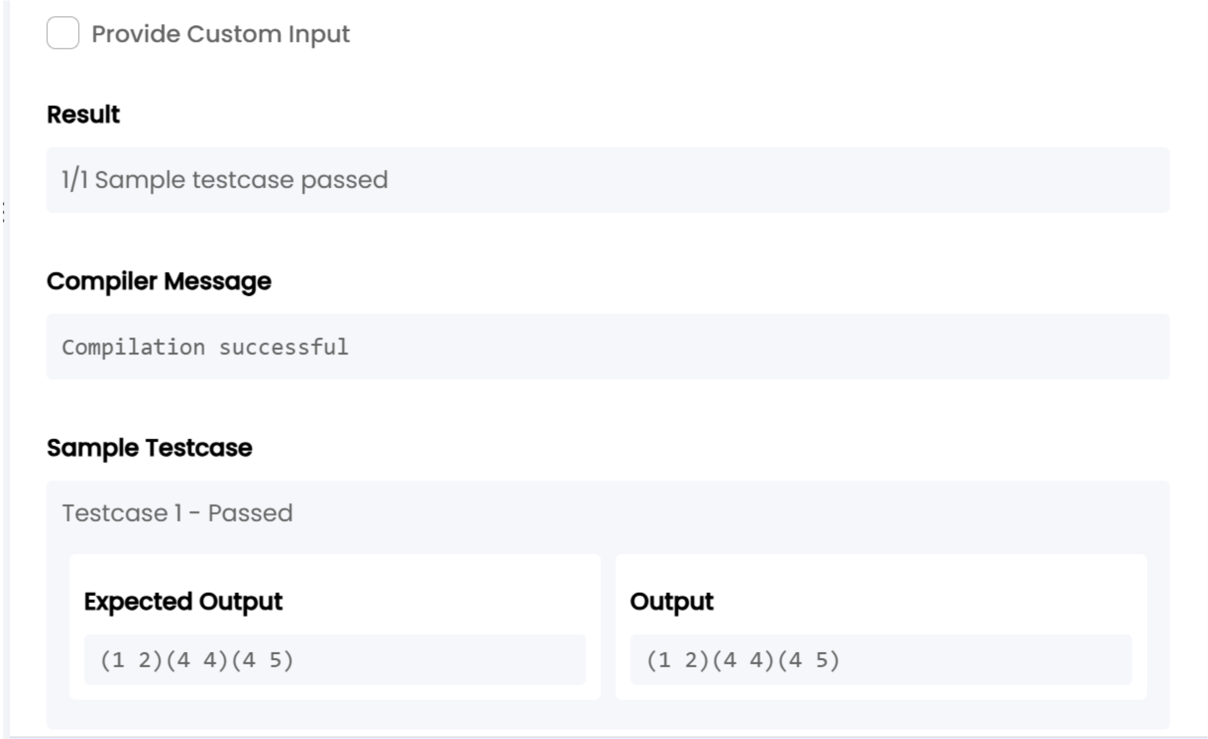
```
7
2 3 6 4 9 0 11
9
```

Output 1 :

```
(1 2)(4 4)(4 5)
```

```
import java.util.*;

public class Main {

  public static void main(String[] args) {

    Scanner sc = new Scanner(System.in);

    int n = sc.nextInt();

    int[] a = new int[n];
```

```
        for(int i=0;i<n;i++)
            a[i]=sc.nextInt();
        int x = sc.nextInt();
        for(int i=0;i<n;i++){
            int sum=0;
            for(int j=i;j<n;j++){
                sum+=a[j];
                if(sum==x)
                    System.out.print("("+i+" "+j+")");
            }
        }
    }
}
```



□ Provide Custom Input

Result

1/1 Sample testcase passed

Compiler Message

```
Compilation successful
```

Sample Testcase

Testcase 1 - Passed

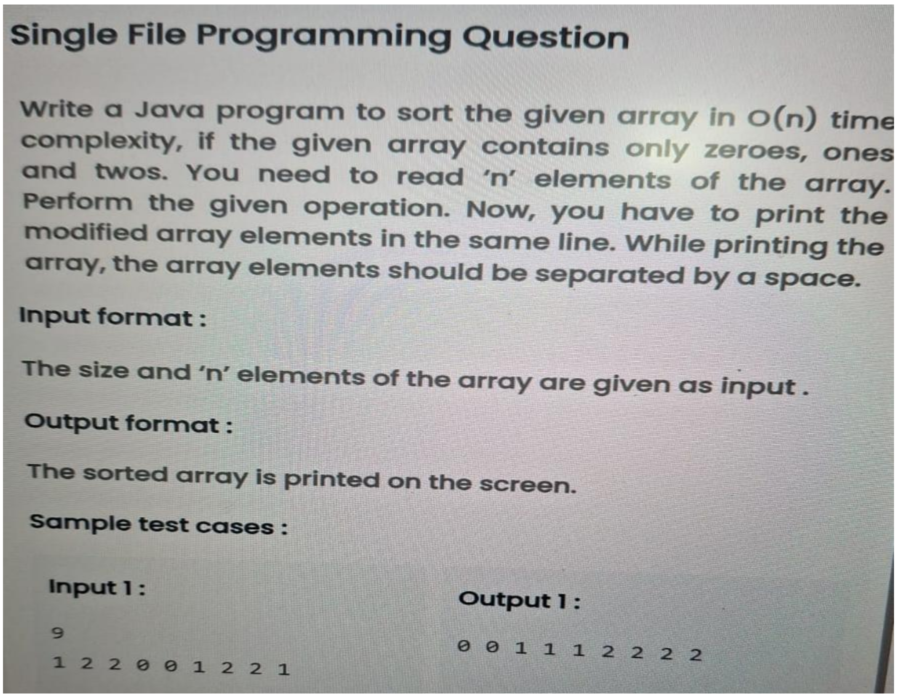
Expected Output

```
(1 2)(4 4)(4 5)
```

Output

```
(1 2)(4 4)(4 5)
```

TASK20:



Single File Programming Question

Write a Java program to sort the given array in O(n) time complexity, if the given array contains only zeroes, ones and twos. You need to read 'n' elements of the array. Perform the given operation. Now, you have to print the modified array elements in the same line. While printing the array, the array elements should be separated by a space.

Input format :

The size and 'n' elements of the array are given as input .

Output format :

The sorted array is printed on the screen.

Sample test cases :

Input 1 :

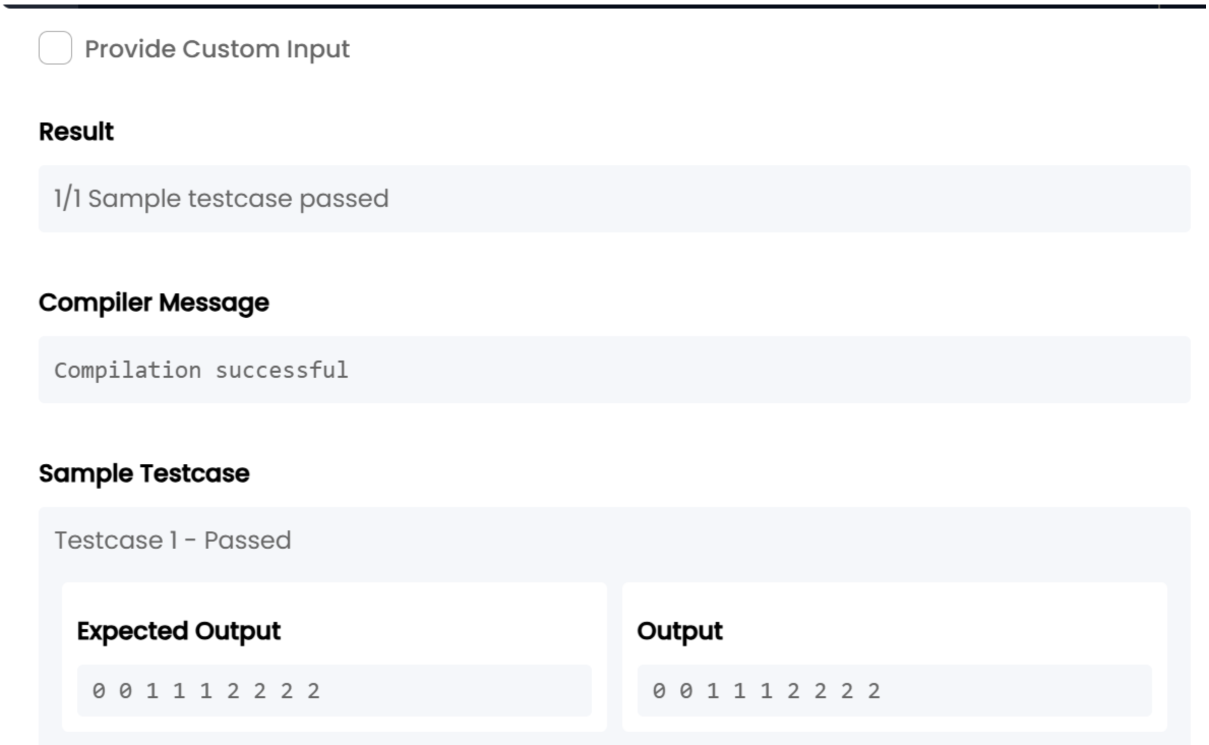
```
9
1 2 2 0 0 1 2 2 1
```

Output 1 :

```
0 0 1 1 1 2 2 2 2
```

```
import java.util.*;
public class Main {
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int n = sc.nextInt();
    int[] a = new int[n];
    for(int i=0;i<n;i++)
      a[i] = sc.nextInt();
    int c0=0,c1=0,c2=0;
    for(int i=0;i<n;i++){
      if(a[i]==0) c0++;
      else if(a[i]==1) c1++;
      else c2++;
    }
    for(int i=0;i<c0;i++) System.out.print("0 ");
    for(int i=0;i<c1;i++) System.out.print("1 ");
    for(int i=0;i<c2;i++) System.out.print("2 ");
  }
```

}



□ Provide Custom Input

Result

1/1 Sample testcase passed

Compiler Message

```
Compilation successful
```

Sample Testcase

Testcase 1 - Passed

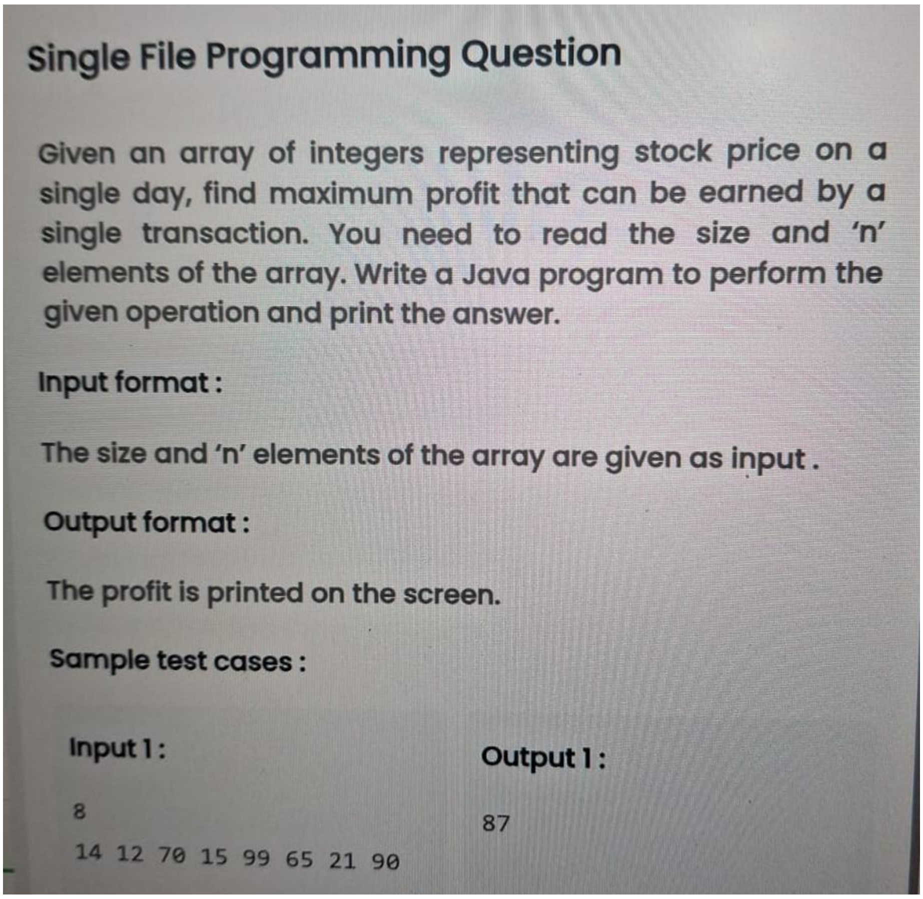
Expected Output

```
0 0 1 1 1 2 2 2 2
```

Output

```
0 0 1 1 1 2 2 2 2
```

TASK21:



Single File Programming Question

Given an array of integers representing stock price on a single day, find maximum profit that can be earned by a single transaction. You need to read the size and 'n' elements of the array. Write a Java program to perform the given operation and print the answer.

Input format :

The size and 'n' elements of the array are given as input .

Output format :

The profit is printed on the screen.

Sample test cases :

Input 1 :

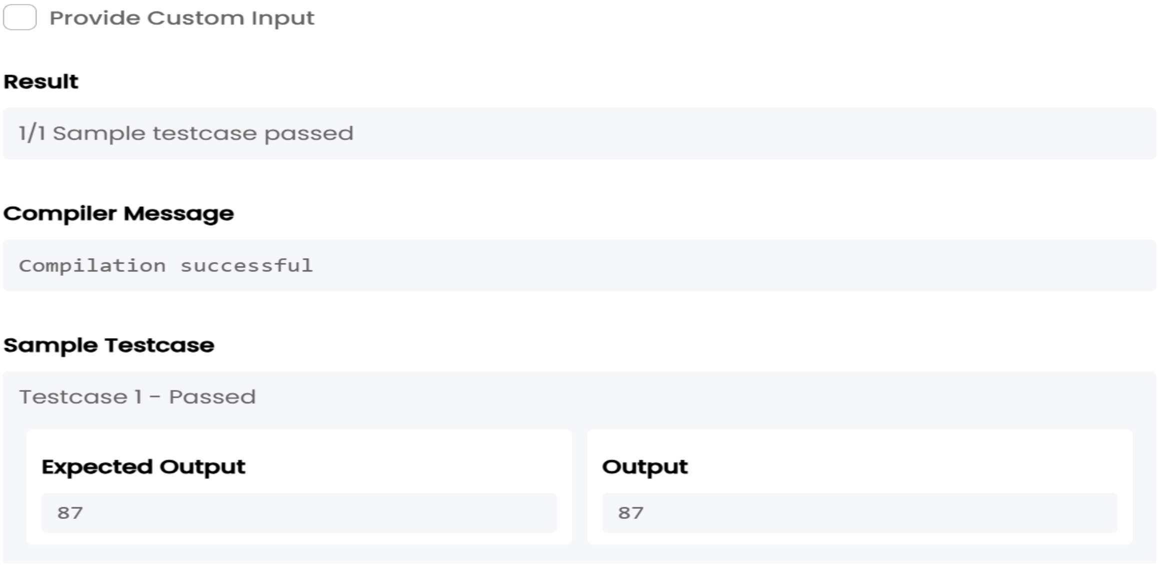
```
8
14 12 70 15 99 65 21 90
```

Output 1 :

```
87
```

import java.util.*;

```
public class Main {
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int n = sc.nextInt();
        int min = Integer.MAX_VALUE, profit = 0;
        for(int i = 0; i < n; i++) {
      int price = sc.nextInt();
      if(price < min)
        min = price;
      else if(price - min > profit)
        profit = price - min;
    }
        System.out.print(profit);
  }
}
```



☐ Provide Custom Input

Result

1/1 Sample testcase passed

Compiler Message

```
Compilation successful
```

Sample Testcase

Testcase 1 - Passed

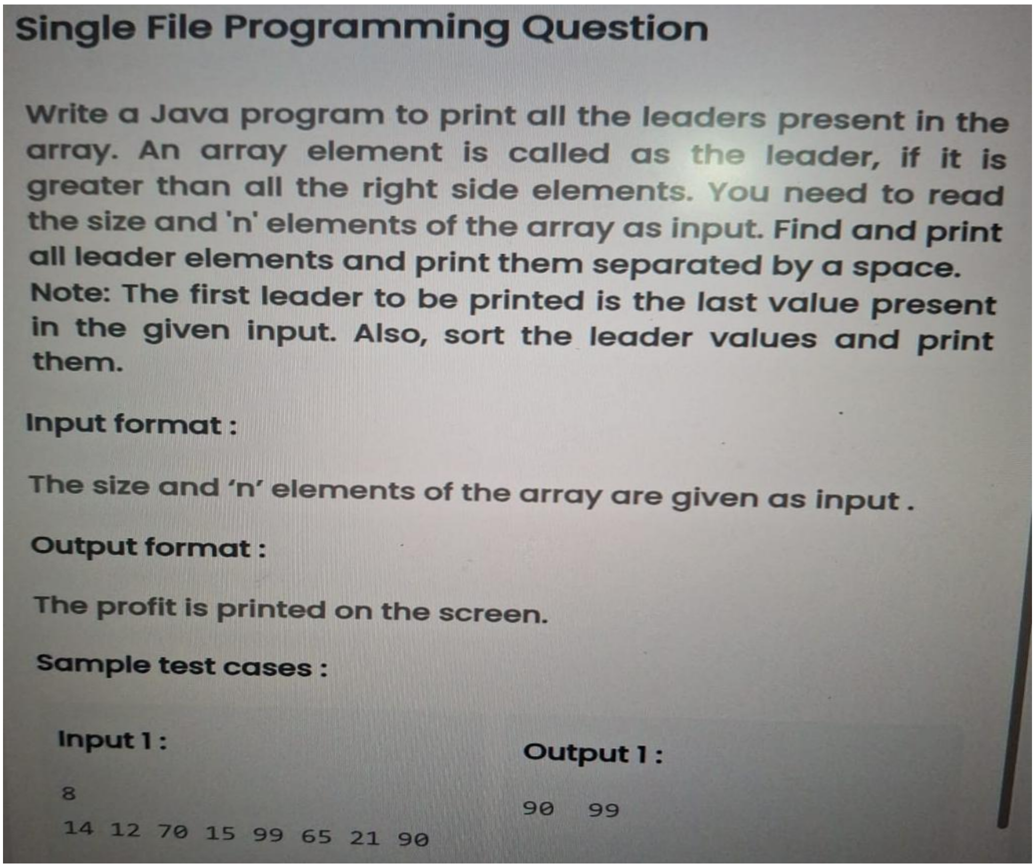
Expected Output

```
87
```

Output

```
87
```

TASK22:



Single File Programming Question

Write a Java program to print all the leaders present in the array. An array element is called as the leader, if it is greater than all the right side elements. You need to read the size and 'n' elements of the array as input. Find and print all leader elements and print them separated by a space. Note: The first leader to be printed is the last value present in the given input. Also, sort the leader values and print them.

Input format :

The size and 'n' elements of the array are given as input .

Output format :

The profit is printed on the screen.

Sample test cases :

Input 1 :

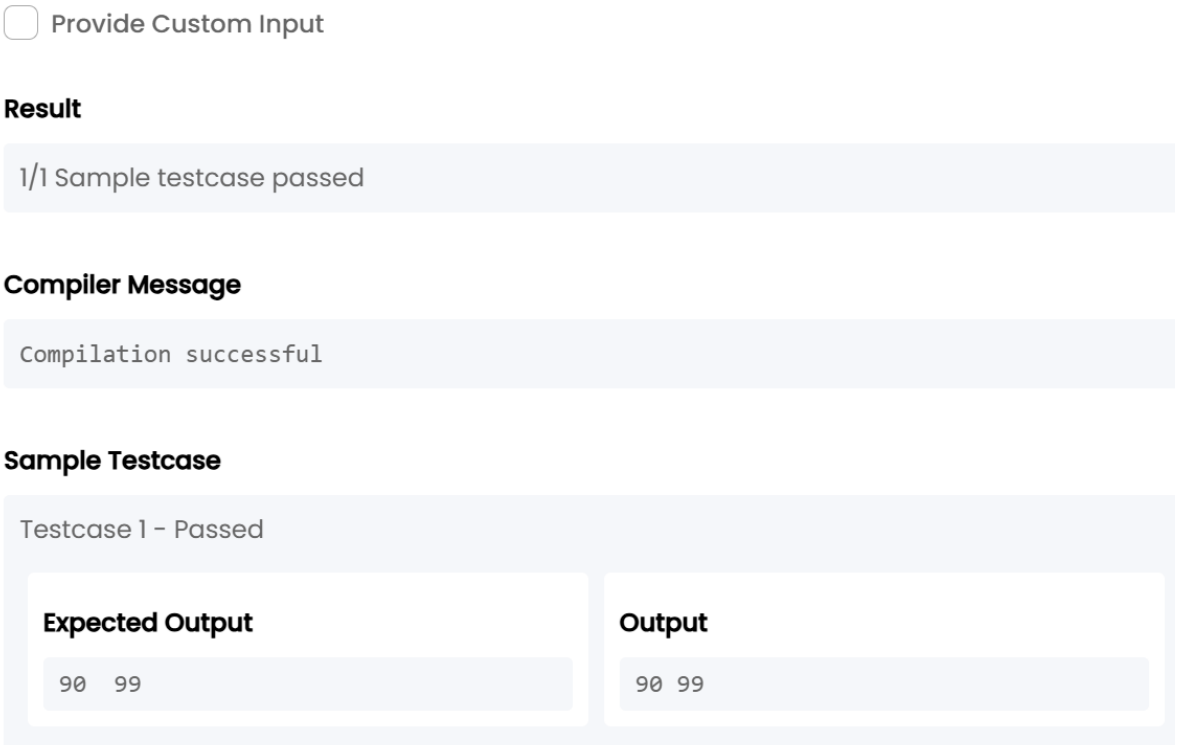
```
8
14 12 70 15 99 65 21 90
```

Output 1 :

```
90  99
```

```
import java.util.*;
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[] a = new int[n];
        for(int i=0;i<n;i++)
            a[i] = sc.nextInt();
        ArrayList<Integer> list = new ArrayList<>();
        int max = Integer.MIN_VALUE;

        for(int i=n-1;i>=0;i--){
            if(a[i] > max){
                list.add(a[i]);
                max = a[i];
```

```
          }
        }
        Collections.sort(list);
        for(int i=0;i<list.size();i++)
          System.out.print(list.get(i)+" ");
      }
    }
```



☐ Provide Custom Input

Result

1/1 Sample testcase passed

Compiler Message

```
Compilation successful
```

Sample Testcase

Testcase 1 - Passed

Expected Output

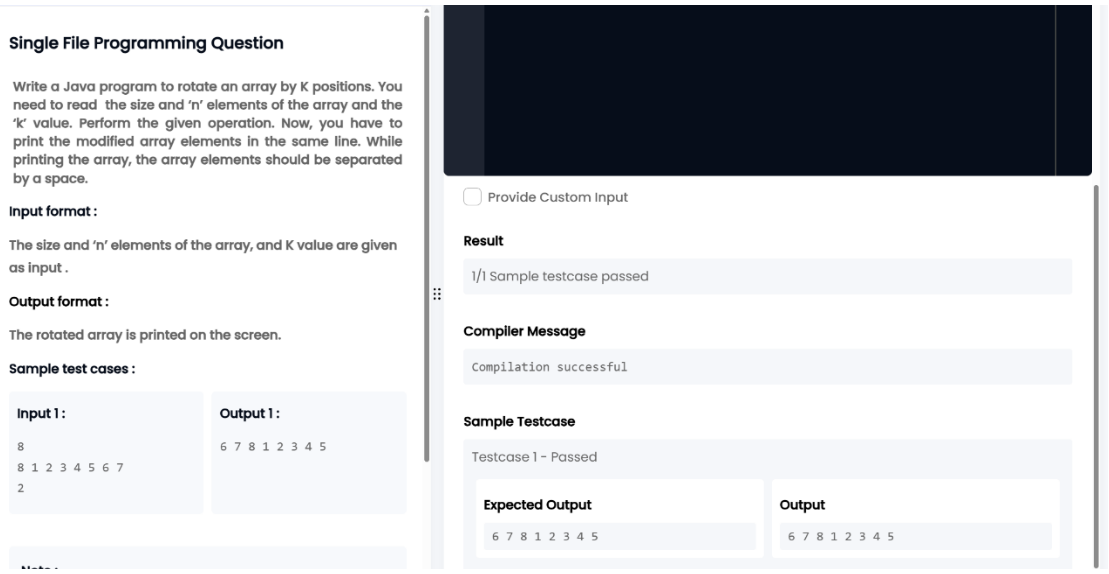
```
90  99
```

Output

```
90 99
```

5. Data structure 4_Search & Sorting_Chapter 3

TASK23:



Single File Programming Question

Write a Java program to rotate an array by K positions. You need to read the size and 'n' elements of the array and the 'k' value. Perform the given operation. Now, you have to print the modified array elements in the same line. While printing the array, the array elements should be separated by a space.

Input format :

The size and 'n' elements of the array, and K value are given as input .

Output format :

The rotated array is printed on the screen.

Sample test cases :

Input 1 :

```
8
8 1 2 3 4 5 6 7
2
```

Output 1 :

```
6 7 8 1 2 3 4 5
```

Note :

□ Provide Custom Input

Result

1/1 Sample testcase passed

Compiler Message

```
Compilation successful
```

Sample Testcase

Testcase 1 - Passed

Expected Output

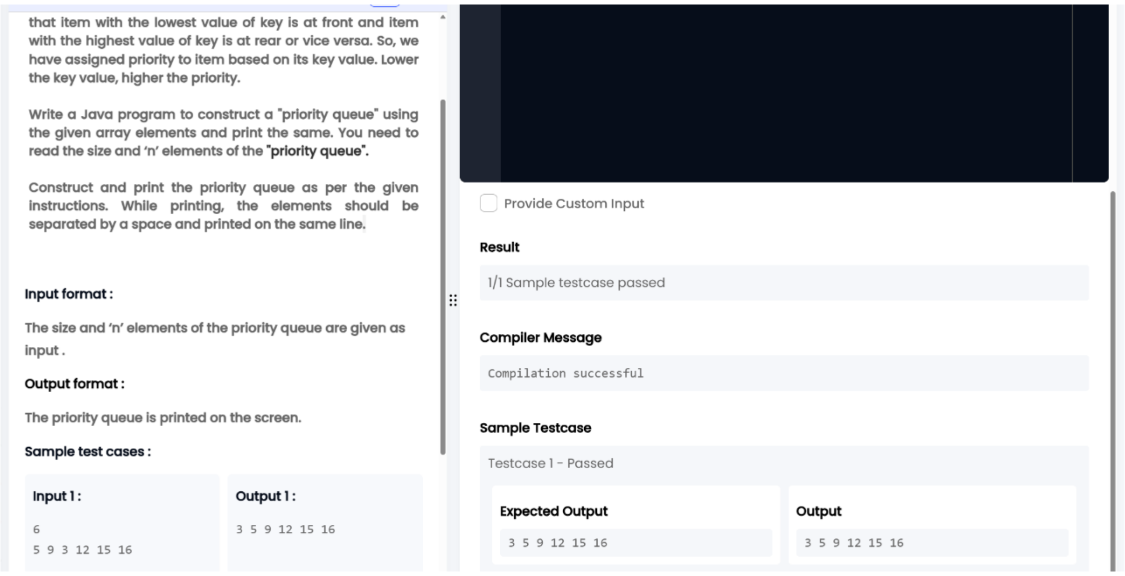
```
6 7 8 1 2 3 4 5
```

Output

```
6 7 8 1 2 3 4 5
```

```
import java.util.*;
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[] a = new int[n];
        for(int i=0;i<n;i++)
            a[i] = sc.nextInt();
        int k = sc.nextInt();
        k = k % n;
        for(int i=n-k;i<n;i++)
            System.out.print(a[i]+" ");
        for(int i=0;i<n-k;i++)
            System.out.print(a[i]+" ");
    }
}
```

TASK24:



that item with the lowest value of key is at front and item with the highest value of key is at rear or vice versa. So, we have assigned priority to item based on its key value. Lower the key value, higher the priority.

Write a Java program to construct a "priority queue" using the given array elements and print the same. You need to read the size and 'n' elements of the **"priority queue"**.

Construct and print the priority queue as per the given instructions. While printing, the elements should be separated by a space and printed on the same line.

Input format :

The size and 'n' elements of the priority queue are given as input .

Output format :

The priority queue is printed on the screen.

Sample test cases :

Input 1 :

```
6
5 9 3 12 15 16
```

Output 1 :

```
3 5 9 12 15 16
```

Result

1/1 Sample testcase passed

Compiler Message

```
Compilation successful
```

Sample Testcase

Testcase 1 - Passed

Expected Output

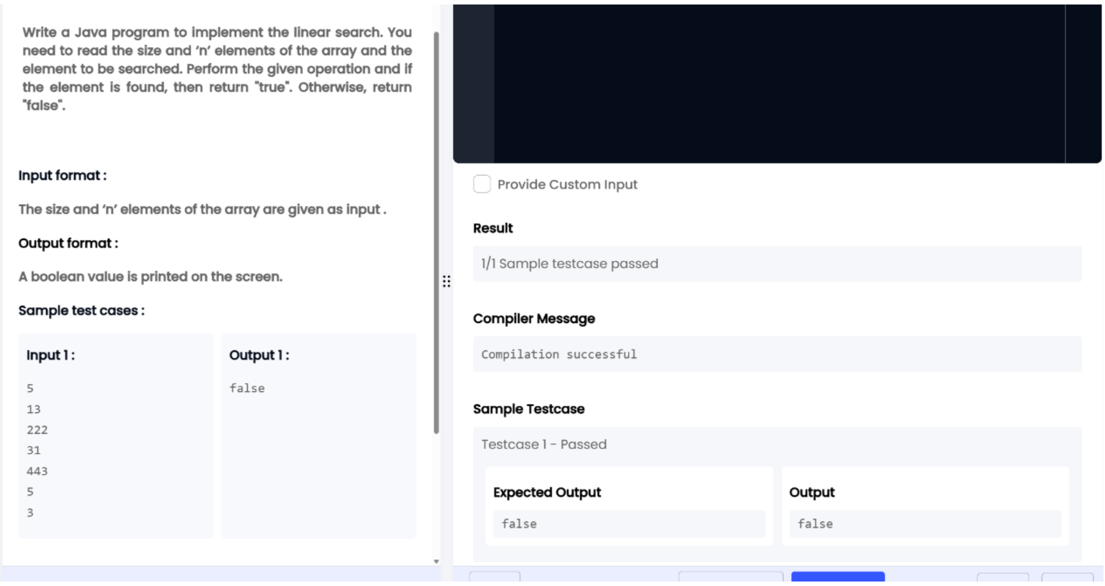
```
3 5 9 12 15 16
```

Output

```
3 5 9 12 15 16
```

```
import java.util.*;
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        PriorityQueue<Integer> pq = new PriorityQueue<>();
        for(int i=0;i<n;i++)
            pq.add(sc.nextInt());
        while(!pq.isEmpty())
            System.out.print(pq.poll()+" ");
    }
}
```

TASK25:



Write a Java program to implement the linear search. You need to read the size and 'n' elements of the array and the element to be searched. Perform the given operation and if the element is found, then return "true". Otherwise, return "false".

Input format :

The size and 'n' elements of the array are given as input .

Output format :

A boolean value is printed on the screen.

Sample test cases :

Input 1 :	Output 1 :
5 13 222 31 443 5 3	false

☐ Provide Custom Input

Result

1/1 Sample testcase passed

Compiler Message

```
Compilation successful
```

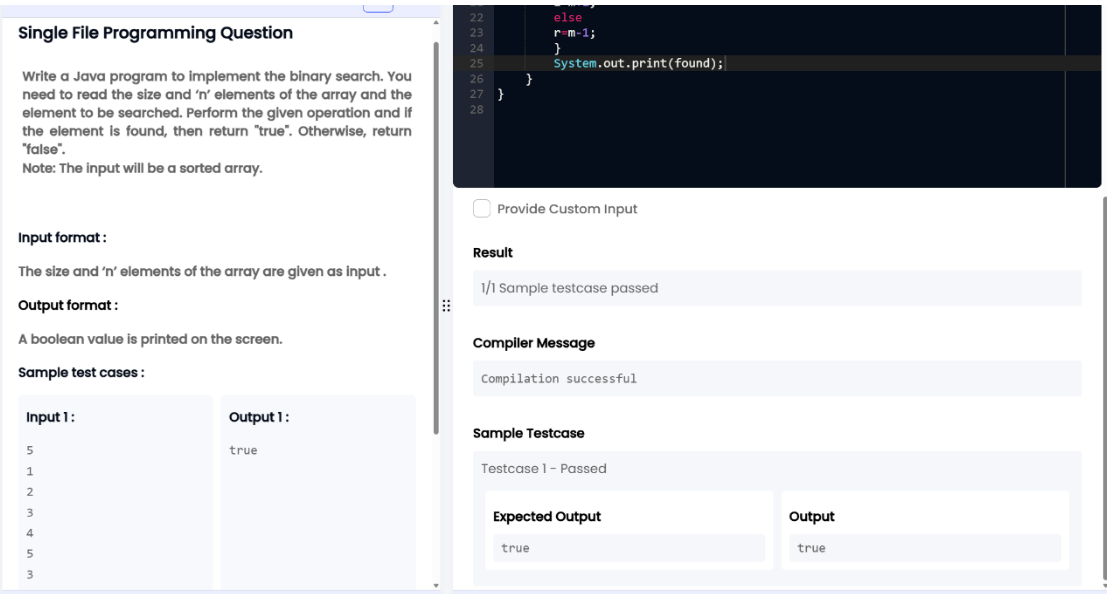
Sample Testcase

Testcase 1 - Passed

Expected Output	Output
false	false

```
import java.util.*;

public class Main {

  public static void main(String[] args) {

    Scanner sc = new Scanner(System.in);

    int n = sc.nextInt();

    int[] a = new int[n];

    for(int i=0;i<n;i++)

      a[i] = sc.nextInt();

    int key = sc.nextInt();

    boolean found = false;

    for(int i=0;i<n;i++)

      if(a[i] == key)

        found = true;

    System.out.print(found);

  }

}
```

TASK26:



Single File Programming Question

Write a Java program to implement the binary search. You need to read the size and 'n' elements of the array and the element to be searched. Perform the given operation and if the element is found, then return "true". Otherwise, return "false".
Note: The input will be a sorted array.

Input format :

The size and 'n' elements of the array are given as input .

Output format :

A boolean value is printed on the screen.

Sample test cases :

Input 1 :	Output 1 :
5 1 2 3 4 5 3	true

```
        else
        r=m-1;
        }
        System.out.print(found);
    }
}
```

□ Provide Custom Input

Result

1/1 Sample testcase passed

Compiler Message

```
Compilation successful
```

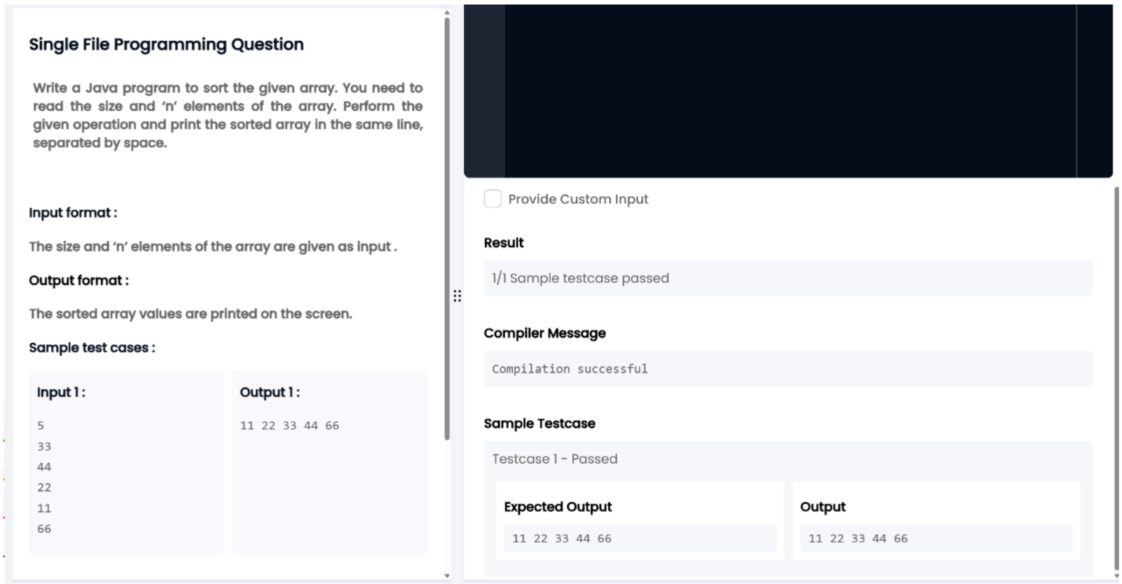
Sample Testcase

Testcase 1 - Passed

Expected Output	Output
true	true

```
import java.util.*;

public class Main {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int n = sc.nextInt();

        int[] a = new int[n];

        for(int i=0;i<n;i++)

            a[i] = sc.nextInt();

        int key = sc.nextInt();

        int l=0, r=n-1;

        boolean found=false;

        while(l<=r){

            int m=(l+r)/2;

            if(a[m]==key){

                found=true;

                break;

            }

            else if(a[m]<key)

                l=m+1;

            else
            r=m-1;
        }
        System.out.print(found);
    }
}
```

TASK27:



Single File Programming Question

Write a Java program to sort the given array. You need to read the size and 'n' elements of the array. Perform the given operation and print the sorted array in the same line, separated by space.

Input format :

The size and 'n' elements of the array are given as input .

Output format :

The sorted array values are printed on the screen.

Sample test cases :

Input 1 :

```
5
33
44
22
11
66
```

Output 1 :

```
11 22 33 44 66
```

□ Provide Custom Input

Result

1/1 Sample testcase passed

Compiler Message

```
Compilation successful
```

Sample Testcase

Testcase 1 - Passed

Expected Output

```
11 22 33 44 66
```

Output

```
11 22 33 44 66
```

```
import java.util.*;
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[] a = new int[n];
        for(int i=0;i<n;i++)
            a[i] = sc.nextInt();
        Arrays.sort(a);
        for(int i=0;i<n;i++)
            System.out.print(a[i] + " ");
    }
}
```