Assignment 3 - Arrays

1. Search for an Element

Problem Statement:

Write a program to search for an element in the array. If found, print its index (0-based); otherwise, print "Not found".

Input:

- First line: Integer n (size of the array)
- Second line: n space-separated integers (array elements)
- Third line: Integer x (element to search)

Output:

- Index of the first occurrence of x in the array, or "Not found"

Constraints:

- 1 <= n <= 100
- -10^4 <= arr[i], x <= 10^4

Sample Input:

5

3 8 2 9 6

9

Sample Output:

3

Program:

```
import java.util.Scanner;

public class Main{

    public static void main(String[] args){

        Scanner sc=new Scanner(System.in);

        int n=sc.nextInt();

        int [] arr=new int[n];

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

```
            arr[i]=sc.nextInt();
        }
            int x=sc.nextInt();
            boolean found=false;
              for(int i=0;i<n;i++){
                 if(arr[i]==x){
                 System.out.println(i);
                 found=true;
                 break;
              }
        }
        if(!found){
            System.out.println("Not  Found");
}
}
}
```

output:

```
5
3 8 2 9 6
9
3
```

2. Duplicate Elements

Problem Statement:

Write a program to identify and print all duplicate elements in a 1D array. If no duplicates are found, print "No duplicates".

Input:

- First line: Integer n (number of elements)
- Second line: n space-separated integers

Output:

- All duplicate elements (in any order)
- Or "No duplicates" if all elements are unique

Constraints:

- 1 <= n <= 100
- -10^4 <= arr[i] <= 10^4

Sample Input:

7

5 3 8 5 6 3 2

Sample Output:

5 3

Program:

```
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int n = sc.nextInt();

        int[] arr = new int[n];

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

```
            arr[i] = sc.nextInt();
    }
      boolean hasDuplicates = false;
           boolean[] printed = new boolean[n];
           for (int i = 0; i < n; i++) {
               if (printed[i]) continue;
               for (int j = i + 1; j < n; j++) {
                   if (arr[i] == arr[j]) {
                       if (!printed[i]) {
                           System.out.print(arr[i] + " ");
                           hasDuplicates = true;
                           printed[i] = true;
                       }
                       printed[j] = true;
                   }
               }
           }
      if (!hasDuplicates) {
               System.out.println("No  duplicates");
           }
        }
    }
```

Output:

7
5 3 8 5 6 3 2
5 3

3. Left Rotation by K Position

Problem Statement:
Write a program to perform **left rotation** of a 1D array by **k** position.

Input:

- irst line: Integer n — the size of the array
- Second line: n space-separated integers – the elements of the array
- Third line: Integer k – number of positions to rotate the array to the left

Output:

- **A single line containing the rotated array elements after k left rotations.**

Constraints:

- 1 <= n <= 100
- -10^4 <= arr[i] <= 10^4
- 0 <= k <= 100

Sample Input:

6

1 2 3 4 5 6

2

Sample Output:

3 4 5 6 1 2

Program

```
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int n = sc.nextInt();

        int[] arr = new int[n];

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

```
            arr[i] = sc.nextInt();

        }

            int k = sc.nextInt();

            k = k % n;

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

                System.out.print(arr[i] + " ");

            }

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

                System.out.print(arr[i] + " ");

            }

        }

}
```

Output:

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