```
Code: 1
package Topic_08_RecursionUsingArrays;
import java.util.Scanner;
public class A_DisplayArray {
        public static void main(String[] args) throws Exception {
                // write your code here
                Scanner s = new Scanner(System.in);
                int n = s.nextInt();
                int a[] = new int[n];
                for (int i = 0; i < n; i++) {
                         a[i] = s.nextInt();
                }
                 displayArr(a, a.length-1);
        }
        public static void displayArr(int[] arr, int idx) {
                if (idx == -1) {
                         return;
                System.out.println(arr[idx]);//To Print Straight
                 displayArr(arr, idx - 1);
                System.out.println(arr[idx]);//To Print In reverse
        }
}
```

```
Code: 2
package Topic_08_RecursionUsingArrays;
import java.util.Scanner;
public class B_FindMaxInArray {
        public static void main(String[] args) throws Exception {
                // write your code here
                Scanner s = new Scanner(System.in);
                int n = s.nextInt();
                int a[] = new int[n];
                for (int i = 0; i < n; i++) {
                        a[i] = s.nextInt();
                }
                int max = maxOfArray(a, a.length - 1);
                System.out.println(max);
        }
        public static int maxOfArray(int[] a, int idx) {
                if (idx == 0) {
                         return a[idx];
                int max = maxOfArray(a, idx - 1);
                if (a[idx] > max) {
                         max = a[idx];
                return max;
        }
}
```

```
Code: 3
package Topic_08_RecursionUsingArrays;
import java.util.Scanner;
public class C_FirstIndex {
        public static void main(String[] args) throws Exception {
                 Scanner s = new Scanner(System.in);
                 int n = s.nextInt();
                 int a[] = new int[n];
                 for (int i = 0; i < n; i++) {
                          a[i] = s.nextInt();
                 int x = s.nextInt();
                 int firstIndex = firstIndex(a, 0, x);
                 System.out.println(firstIndex);
        }
        public static int firstIndex(int[] arr, int idx, int x) {
                 if (idx == arr.length) {
                          return -1;
                 if (arr[idx] == x) {
                          return idx;
                 int fi = firstIndex(arr, idx + 1, x);
                 return fi;
        }
}
```

```
Code: 4
package Topic_08_RecursionUsingArrays;
import java.util.Scanner;
public class D_LastIndex {
        public static void main(String[] args) throws Exception {
                 Scanner s = new Scanner(System.in);
                 int n = s.nextInt();
                 int a[] = new int[n];
                 for (int i = 0; i < n; i++) {
                          a[i] = s.nextInt();
                 int x = s.nextInt();
                 int lastIndex = lastIndex(a, a.length - 1, x);
                 System.out.println(lastIndex);
        }
        public static int lastIndex(int[] arr, int idx, int x) {
                 if (idx == -1) {
                          return -1;
                 int li = lastIndex(arr, idx - 1, x);
                 if (arr[idx] == x) {
                          return idx;
                 }
                 return li;
        }
}
```

```
Code: 5
package Topic_08_RecursionUsingArrays;
import java.util.lterator;
import java.util.Scanner;
public class E AllIndicesOfArray {
        public static void main(String[] args) throws Exception {
                 Scanner s = new Scanner(System.in);
                 int n = s.nextInt();
                 int a[] = new int[n];
                 for (int i = 0; i < n; i++) {
                          a[i] = s.nextInt();
                 int x = s.nextInt();
                 int arr[] = allIndices(a, x, 0, 0);
                 for (int i : arr) {
                          System.out.println(i);
                 }
        }
        public static int[] allIndices(int[] arr, int x, int idx, int count) {
                 if (arr.length == idx) {
                          return new int[count];
                 if (arr[idx] == x) {
                          count++;
                 int[] a = allIndices(arr, x, idx + 1, count);
                 if (arr[idx] == x) {
                          a[count - 1] = idx;
                 return a;
        }
}
```