```
public class Sorting {
    // Find first and last positions of an element in a sorted array

    class Solution {
    ArrayList<Long> find(long arr[], int n, int x) {
        // code here
        ArrayList<Long> ls = new ArrayList<>();
        int first = -1;
        int second = -1;
        for (int i = 0; i < n; i++) {
            if (arr[i] != x) {
                continue;
            } else {
                if (first == -1) {
                    first = i;
                }
                second = i;
            }
        }

        int s = 0, e = n - 1;
        while (s < e) {
            int mid = (s + e) / 2;
            if (arr[mid] == x) {
                first = mid;
                e = mid - 1;
            } else {
                s = mid + 1;
            }
        }
        int s1 = 0, e1 = n - 1;
        while (s1 < e1) {
            int mid = (s1 + e1) / 2;
            if (arr[mid] == x) {
                second = mid;
                s1 = mid + 1;

            } else {
                e1 = mid - 1;
            }
        }
        long f = (long) first;
        long s2 = (long) second;
        ls.add(f);
        ls.add(s2);
        return ls;

    }
}
```

```
// Find a Fixed Point (Value equal to index) in a given array

// Search in a rotated sorted array
// Square root of an integer
// Maximum and minimum of an array using minimum number of comparisons
// Optimum location of point to minimize total distance
// Find the repeating and the missing
class Solution {
    int[] findTwoElement(int arr[], int n) {
        // code here
        int sum = 0, sum_sq = 0;
        for (int i = 0; i < n; i++) {
            sum += arr[i];
            sum_sq += arr[i] * arr[i];
        }
        int req_sum = n * (n + 1) / 2;
        int req_sum_sq = (n * (n + 1) * (2 * n + 1)) / 6;
        int a = Math.abs(sum - req_sum);
        int b = Math.abs(sum_sq - req_sum_sq);
        int num1 = 0;
        int num2 = 0;
        int eq2 = b / a;
        int eq1 = a;
        num1 = Math.abs((eq1 + eq2) / 2);
        num2 = Math.abs((eq1 - eq2) / 2);
        return new int[] { num1, num2 };
    }
    class Sol{

    }
}
// Find majority element
// Searching in an array where adjacent differ by at most k
// Find a pair with a given difference
// Find four elements that sum to a given value
// Maximum sum such that no 2 elements are adjacent
// Count triplet with sum smaller than a given value
// Merge 2 sorted arrays
// Product array Puzzle
// Sort array according to count of set bits
// Minimum no. of swaps required to sort the array
// Bishu and Soldiers
// Rasta and Kheshtak
// Kth smallest number again
// Find pivot element in a sorted array
// K-th Element of Two Sorted Arrays
// Aggressive cows
// Book Allocation Problem
```

```
    // EKOSPOJ:
    // Job Scheduling Algo
    // Missing Number in AP
    // Smallest number with atleast n trailing zeroes in factorial
    // Painters Partition Problem:
    // ROTI-Prata SPOJ
    // DoubleHelix SPOJ
    // Subset Sums
    // Find the inversion count
    // Implement Merge-sort in-place
    // Partitioning and Sorting Arrays with Many Repeated Entries

}
```