LAB # 06

Searching in a Linear Array

OBJECTIVE: To find an element in linear array using Linear Search and Binary Search.

LAB TASKS

1.Declare an array of size 10 to store account balances. Initialize with values 0 to 1000000. Check all array if any value is less than 10000.

Code:

```
public class Main {
    public static void main(String[] args) {
        int[] balances = {12000, 5000, 8000, 30000, 10000, 15000, 2000, 50000, 7000, 1000000};
        for (int i = 0; i < balances.length; i++) {
            if (balances[i] < 10000) {
                System.out.println("Account No. " + (i + 1) + " Low Balance");
            }
        }
    }
}
```

Output:

```
Account No. 2 Low Balance
Account No. 3 Low Balance
Account No. 7 Low Balance
Account No. 9 Low Balance
```

2.Write a program to search in array using Array built-in class.

Code:

```
import java.util.Arrays;
public class Main {
    public static void main(String[] args) {
        int[] arr = {15, 30, 7, 10, 22, 18};
        Arrays.sort(arr);
        int index = Arrays.binarySearch(arr, 10);
        System.out.println(index);
    }
}
```

Output:

```
1
```

3. Given an unsorted array arr of integers, find the smallest positive integer that is missing from the array. You need to implement this using binary search. The array can contain both negative numbers and positive numbers, and you can assume that the array does not have duplicates.

Code:

```
import java.util.Arrays;
public class Main {
    public static void main(String[] args) {
        int[] arr = {-1, 0, 1, 2, 3, 5};
        Arrays.sort(arr);
        int missing = 1;
        for (int i = 0; i < arr.length; i++) {
            if (arr[i] == missing) {
                missing++;
            }
        }
        System.out.println(missing);
    }
}
```

Output:

```
4
```

4.You are given a sorted array arr[] and a target element target. Your task is to find the first occurrence of the target in the array using binary search. If the target is not found, return -1.
You are given a sorted array arr[] and a target element target. Your task is to find the first occurrence of the target in the array using binary search. If the target is not found, return -1.

Code:

```
public class Main {
    public static int binarySearch(int[] arr, int target) {
        int low = 0, high = arr.length - 1;
        int result = -1;
        while (low <= high) {
            int mid = (low + high) / 2;
            if (arr[mid] == target) {
                result = mid;
                high = mid - 1;
            } else if (arr[mid] < target) {
                low = mid + 1;
            } else {
                high = mid - 1;
            }
        }
        return result;
    }
    public static void main(String[] args) {
        int[] arr = {1, 2, 2, 3, 4, 5};
        System.out.println(binarySearch(arr, 2));
    }
}
```

Output: 1

Home Tasks

1.Write a program initializing array of size 20 and search an element using binary search.

Code:

```
import java.util.Arrays;
public class Main {
    public static void main(String[] args) {
        int[] arr = {25, 15, 18, 19, 20, 22, 5, 6, 7, 8, 9, 10, 3, 2, 1, 14, 13, 17, 11, 12};
        Arrays.sort(arr);
        int target = 10;
        int low = 0, high = arr.length - 1, mid, loc = -1;
        while (low <= high) {
            mid = (low + high) / 2;
            if (arr[mid] == target) {
                loc = mid;
                break;
            } else if (arr[mid] < target) {
                low = mid + 1;
            } else {
                high = mid - 1;
            }
        }
        System.out.println(loc);
    }
}
```

Output: 8

2. Write a function called occurrences that, given an array of numbers A, prints all the distinct values in A each followed by its number of occurrences.

Code:

```
public class Main {
    public static void main(String[] args) {
        int[] arr = {28, 1, 0, 1, 0, 3, 4, 0, 0, 3};
        for (int i = 0; i < arr.length; i++) {
            if (arr[i] == -1) continue;
            int count = 1;
            for (int j = i + 1; j < arr.length; j++) {
                if (arr[i] == arr[j]) {
                    count++;
                    arr[j] = -1;
                }
            }
            System.out.println(arr[i] + " " + count);
        }
    }
}
```

Output:

```
28 1
1 2
0 4
3 2
4 1
```