Experiment 1.4

Student Name: Nishant Kumar mehta **UID: 21BCS3402**
Branch: CSE **Section/Group: IOT - 602**
Semester: 5th **Dateof Performance: 01/09/23**
Subject Name: Advance Programming Lab **Subject Code: 21CSP - 314**

1. Aim: Sorting and Searching: Implement the concept of Searching and Sorting techniques.

2. Objective:

- HackerLand National Bank has a simple policy for warning clients about possible fraudulent account activity. If the amount spent by a client on a particular day is greater than or equal to 2x the client's median spending for a trailing number of days, they send the client a notification about potential fraud. The bank doesn't send the client any notifications until they have at least that trailing number of prior days' transaction data.Given the number of trailing days and a client's total daily expenditures for a period of n days, determine the number of times the client will receive a notification over all n days.
- Given two arrays of integers, find which elements in the second array are missing from the first array.

3. Program and output:

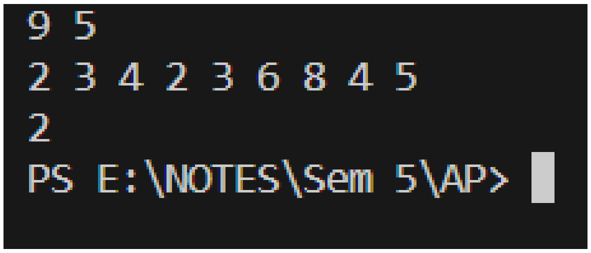
```
import java.util.*;


public class Exp4 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int d = sc.nextInt();
        boolean isEven = true;
        if (d % 2 != 0)
            isEven = false;
```

```
int warning_count = 0;
int[] expenditure = new int[n];
for (int i = 0; i < n; i++) {
    expenditure[i] = sc.nextInt();
}
int[] data = new int[201];
for (int i = 0; i < d; i++) {
    data[expenditure[i]]++;
}
for (int i = d; i < expenditure.length; i++) {

    double median_double = 0;
    if (isEven) {
        int m1 = -1;
        int m2 = -1;
        int count = 0;
        for (int j = 0; j < data.length; j++) {
            count += data[j];
            if (m1 < 0 && count >= d / 2) {
                m1 = j;
            }
            if (m2 < 0 && count >= d / 2 + 1) {
                m2 = j;
                break;
            }
        }
```

```
                median_double = m1 + m2;
            } else {
                int count = 0;
                for (int j = 0; j < data.length; j++) {
                    count += data[j];
                    if (count > d / 2) {
                        median_double = j * 2;
                        break;
                    }
                }
            }
            if (expenditure[i] >= median_double) {
                warning_count++;
            }
            data[expenditure[i]]++;
            data[expenditure[i - d]]--;
        }
        System.out.println(warning_count);
        sc.close();
    }
}
```

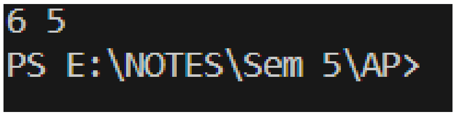


```
9 5
2 3 4 2 3 6 8 4 5
2
PS E:\NOTES\Sem 5\AP>
```

2.

```
 import java.util.HashSet;
import java.util.Set;

public class Exp4Q{
    static void findMissing(int a[], int b[],
                    int n, int m)
    {
        HashSet<Integer> s = new HashSet<>();
        for (int i = 0; i < m; i++)
            s.add(b[i]);
        for (int i = 0; i < n; i++)
            if (!s.contains(a[i]))
                System.out.print(a[i] + " ");
    }
    public static void main(String []args){
        int a[] = { 1, 2, 6, 3, 4, 5 };
        int b[] = { 2, 4, 3, 1, 0 };
        int n = a.length;
        int m = b.length;
        findMissing(a, b, n, m);
    }
}
```



```
6 5
PS E:\NOTES\Sem 5\AP>
```