TASK 3

AIM: To compare the ratings of Alice and Bob in three categories and calculate their comparison scores based on given rules.

ALGORITHM:

1. Start
2. Read three integers into list a (Alice's ratings).
3. Read three integers into list b (Bob's ratings).
4. Initialize two variables:
   - alice = 0
   - bob = 0
5. For i from 0 to 2:
   - If a[i] > b[i], increment alice
   - Else if a[i] < b[i], increment bob
   - Else, do nothing
6. Store alice and bob scores in a result list.
7. Print the result.
8. Stop

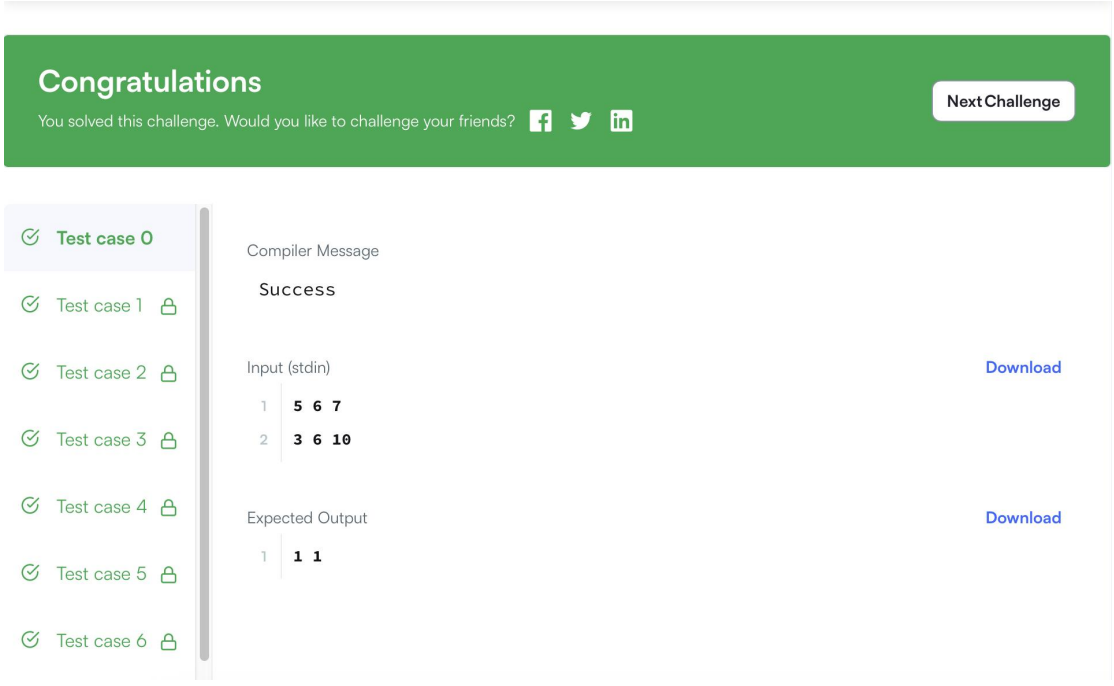
PROGRAM:

```
import java.io.*;
import java.util.*;

class Result {



    public static List<Integer> compareTriplets(List<Integer> a, List<Integer> b) {

        int alice = 0;
        int bob = 0;

        for (int i = 0; i < 3; i++) {
            if (a.get(i) > b.get(i)) {
                alice++;
            } else if (a.get(i) < b.get(i)) {
                bob++;
            }
        }

        List<Integer> result = new ArrayList<>();
        result.add(alice);
        result.add(bob);

        return result;
    }
}

public class Solution {
    public static void main(String[] args) throws IOException {

        BufferedReader bufferedReader = new BufferedReader(new
InputStreamReader(System.in));
        BufferedWriter bufferedWriter = new BufferedWriter(
```

```
            new FileWriter(System.getenv("OUTPUT_PATH")));

        String[] aTemp = bufferedReader.readLine().trim().split(" ");
        List<Integer> a = new ArrayList<>();

        for (int i = 0; i < 3; i++) {
            a.add(Integer.parseInt(aTemp[i]));
        }

        String[] bTemp = bufferedReader.readLine().trim().split(" ");
        List<Integer> b = new ArrayList<>();

        for (int i = 0; i < 3; i++) {
            b.add(Integer.parseInt(bTemp[i]));
        }

        List<Integer> result = Result.compareTriplets(a, b);

        bufferedWriter.write(result.get(0) + " " + result.get(1));
        bufferedWriter.newLine();

        bufferedReader.close();
        bufferedWriter.close();
    }
}import java.io.*;
import java.util.*;

class Result {



    public static List<Integer> compareTriplets(List<Integer> a, List<Integer> b) {

        int alice = 0;
        int bob = 0;

        for (int i = 0; i < 3; i++) {
            if (a.get(i) > b.get(i)) {
                alice++;
            } else if (a.get(i) < b.get(i)) {
                bob++;
            }
        }

        List<Integer> result = new ArrayList<>();
        result.add(alice);
        result.add(bob);

        return result;
    }
}

public class Solution {
    public static void main(String[] args) throws IOException {

        BufferedReader bufferedReader = new BufferedReader(new
InputStreamReader(System.in));
        BufferedWriter bufferedWriter = new BufferedWriter(
```

```
            new FileWriter(System.getenv("OUTPUT_PATH")));

        String[] aTemp = bufferedReader.readLine().trim().split(" ");
        List<Integer> a = new ArrayList<>();

        for (int i = 0; i < 3; i++) {
            a.add(Integer.parseInt(aTemp[i]));
        }

        String[] bTemp = bufferedReader.readLine().trim().split(" ");
        List<Integer> b = new ArrayList<>();

        for (int i = 0; i < 3; i++) {
            b.add(Integer.parseInt(bTemp[i]));
        }

        List<Integer> result = Result.compareTriplets(a, b);

        bufferedWriter.write(result.get(0) + " " + result.get(1));
        bufferedWriter.newLine();

        bufferedReader.close();
        bufferedWriter.close();
    }
}.
```

OUTPUT:



RESULT:The program successfully compares Alice's and Bob's ratings and prints their final scores (Alice first, Bob second)

TASK 4

AIM:To check whether an integer array contains any duplicate elements.

ALGORITHM:

1 Create an empty HashSet.
2 Traverse each element in the array.
3 If the element already exists in the set, return true.
4 Otherwise, add the element to the set.
5 If the loop ends without finding duplicates, return false.

PROGRAM:

```
import java.util.*;

class Solution {
    public boolean containsDuplicate(int[] nums) {

        Set<Integer> seen = new HashSet<>();

        for (int num : nums) {
            if (seen.contains(num)) {
                return true;
            }
            seen.add(num);
        }

        return false;
    }
}
```

OUTPUT:

RESULT:The program successfully detects duplicates in the array and returns true if any element appears more than once; otherwise, it returns false.