Experiment-8

Student_Name: Prince Kumar **Uid: 22BCS13943**
Branch: CSE **Section/Group: 630/A**
Semester: 6th **Date of Performance:16-04-25**
Subject Name: AP Lab **Subject Code: 22CSP-351**

1. **Aim:**

There are n children standing in a line. Each child is assigned a rating value given in the integer array ratings. You are giving candies to these children subject to the following requirements:

- Each child must have at least one candy.
- Children with a higher rating get more candies than their neighbors.

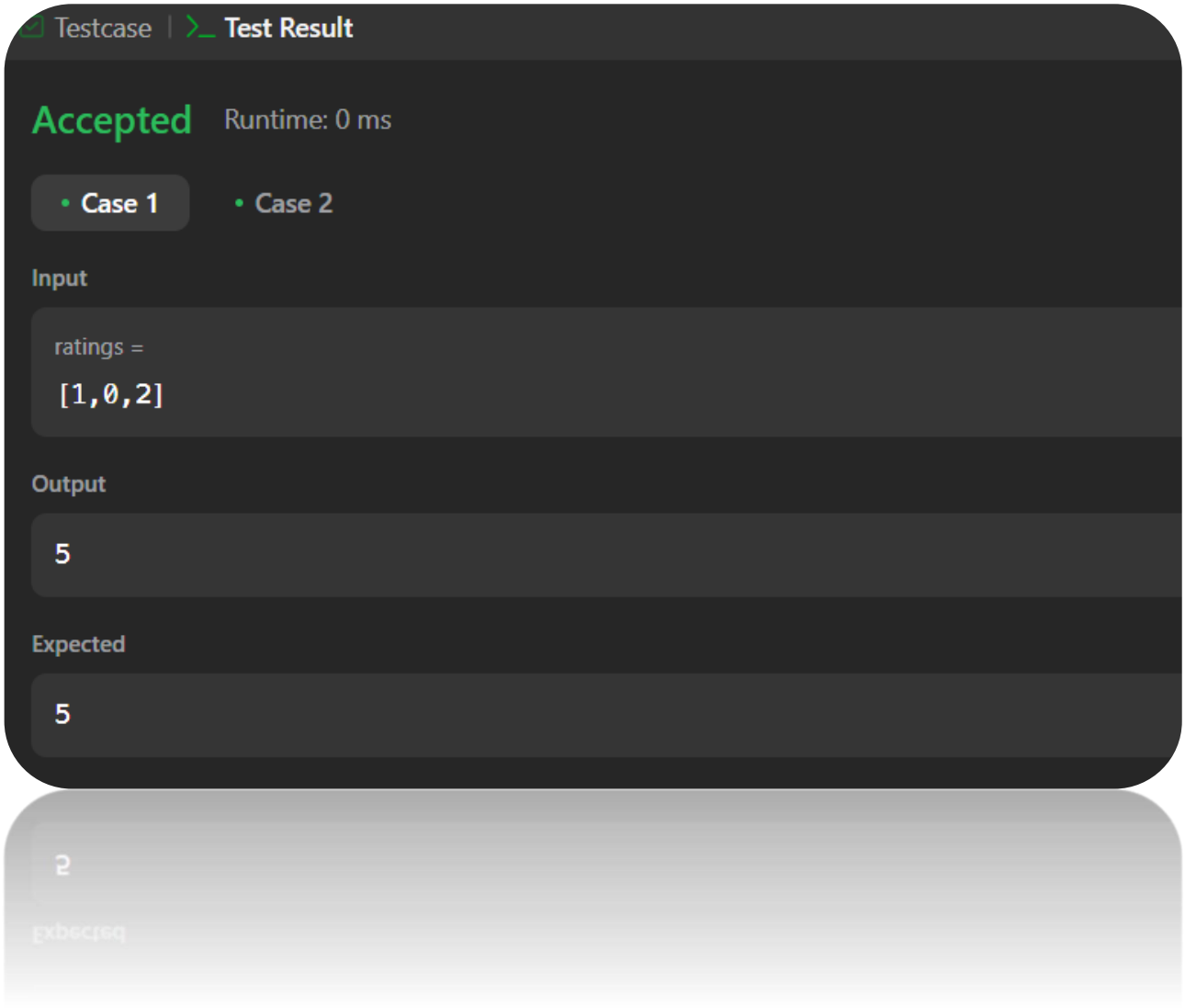
Return the minimum number of candies you need to distribute to the children.

2. Code

```
class Solution {

  public int candy(int[] ratings) {

      int n = ratings.length;

      int[] candies = new int[n];

    for (int i = 0; i < n; i++) {

        candies[i] = 1;

      }
```

```
        for (int i = 1; i < n; i++) {
            if (ratings[i] > ratings[i - 1]) {
                candies[i] = candies[i - 1] + 1;
            }
        }
    for (int i = n - 2; i >= 0; i--) {
            if (ratings[i] > ratings[i + 1]) {
                candies[i] = Math.max(candies[i], candies[i + 1] + 1);
            }
        }   int total = 0;
        for (int c : candies) {
            total += c;
        }return total;
    }
}
```

3. Output:



4. Learning Outcomes

1. Understand how to apply greedy algorithms for optimal resource distribution.
2. Learn to handle edge cases by initializing values with minimum required conditions.
3. Practice two-pass traversal (left-to-right and right-to-left) to enforce conditions from both sides.
4. Understand how to use Math.max() to preserve previously assigned optimal values.

5. Develop skills to convert problem statements into efficient and readable Java code.