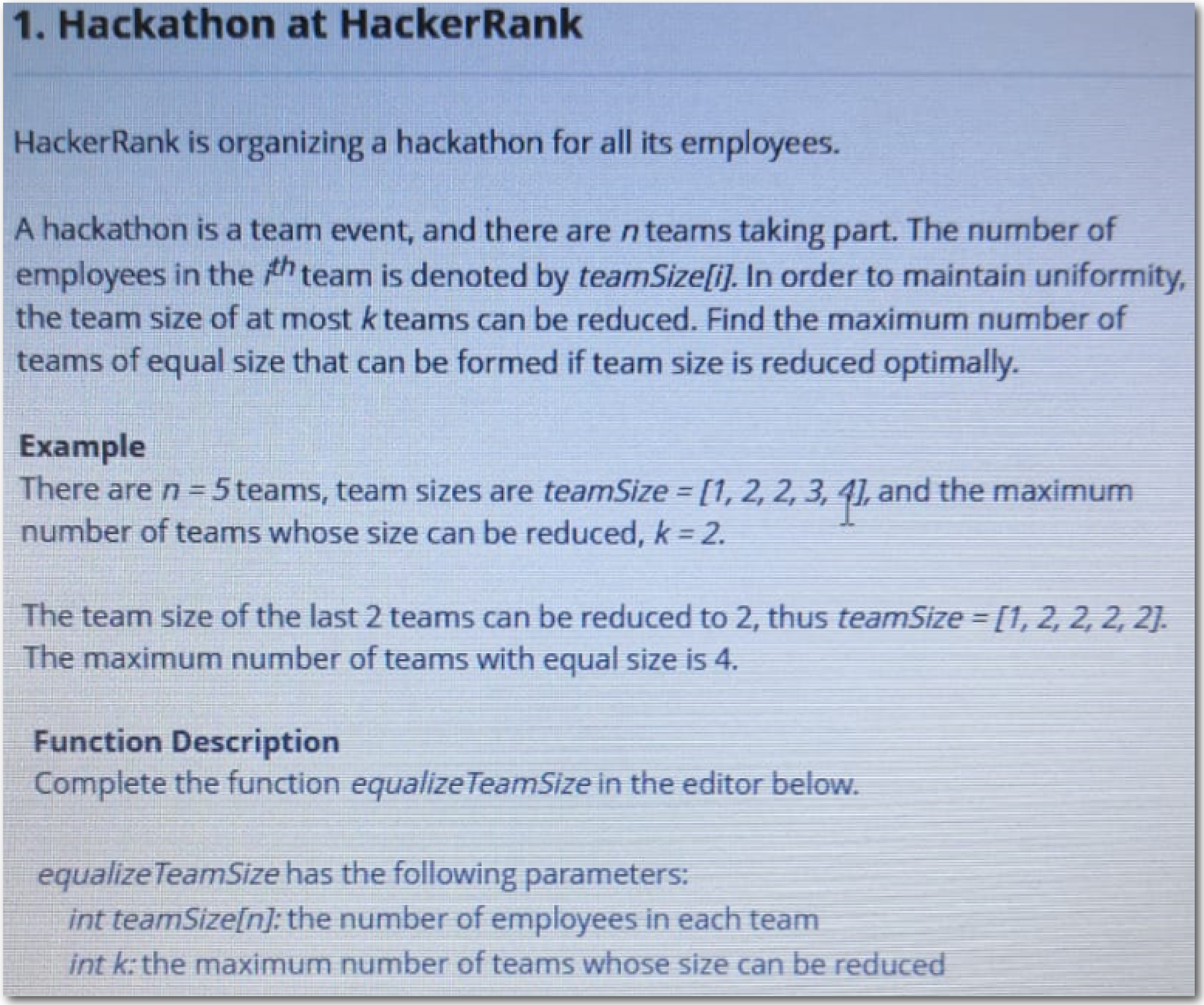


1. Hackathon at HackerRank

HackerRank is organizing a hackathon for all its employees.

A hackathon is a team event, and there are *n* teams taking part. The number of employees in the i^{th} team is denoted by *teamSize[i]*. In order to maintain uniformity, the team size of at most *k* teams can be reduced. Find the maximum number of teams of equal size that can be formed if team size is reduced optimally.

Example

There are *n = 5* teams, team sizes are *teamSize = [1, 2, 2, 3, 4]*, and the maximum number of teams whose size can be reduced, *k = 2*.

The team size of the last 2 teams can be reduced to 2, thus *teamSize = [1, 2, 2, 2, 2]*. The maximum number of teams with equal size is 4.

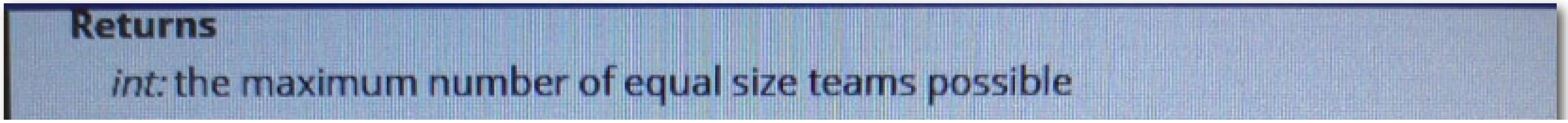
Function Description

Complete the function *equalizeTeamSize* in the editor below.

equalizeTeamSize has the following parameters:

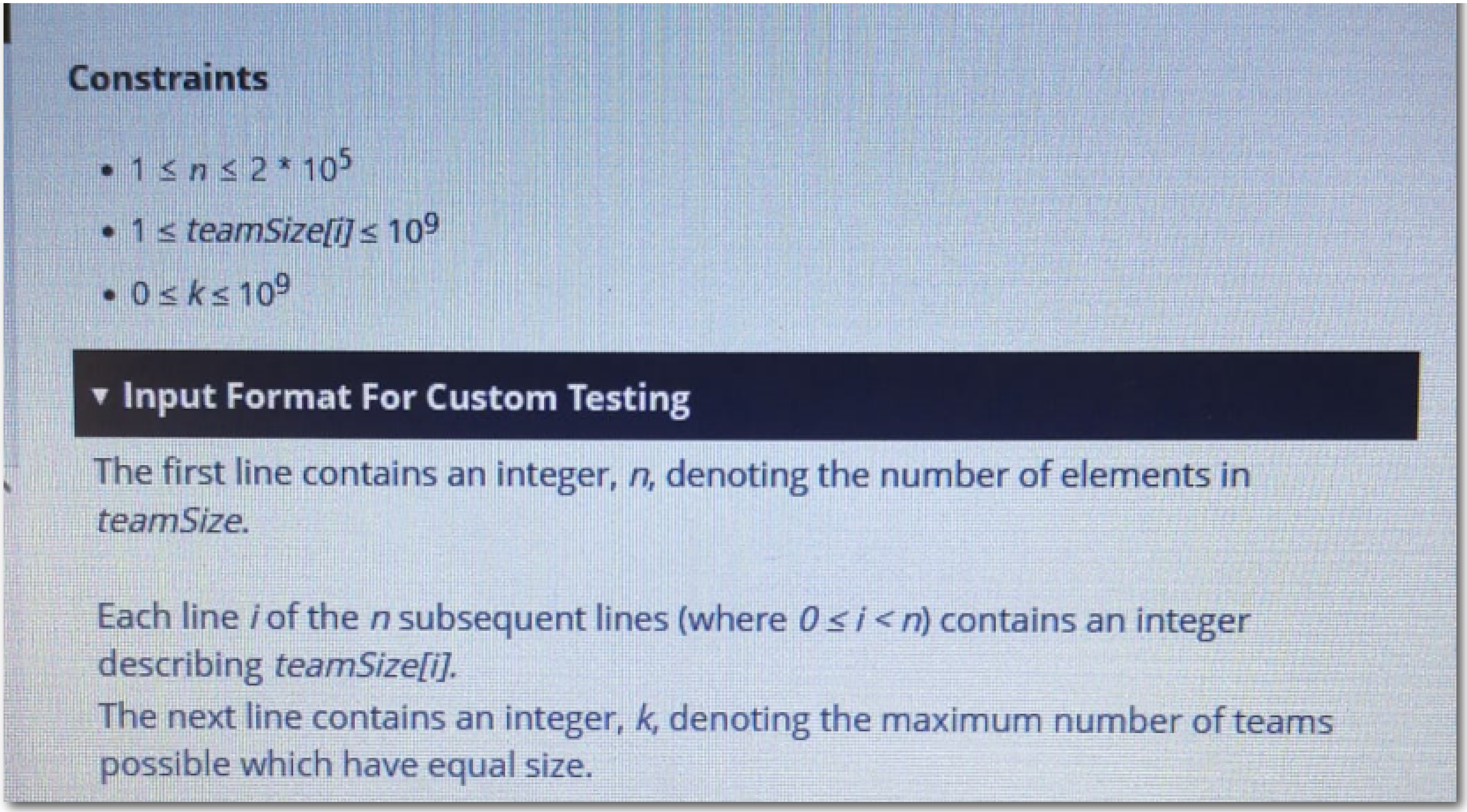
int teamSize[n]: the number of employees in each team

int k: the maximum number of teams whose size can be reduced



Returns

int: the maximum number of equal size teams possible



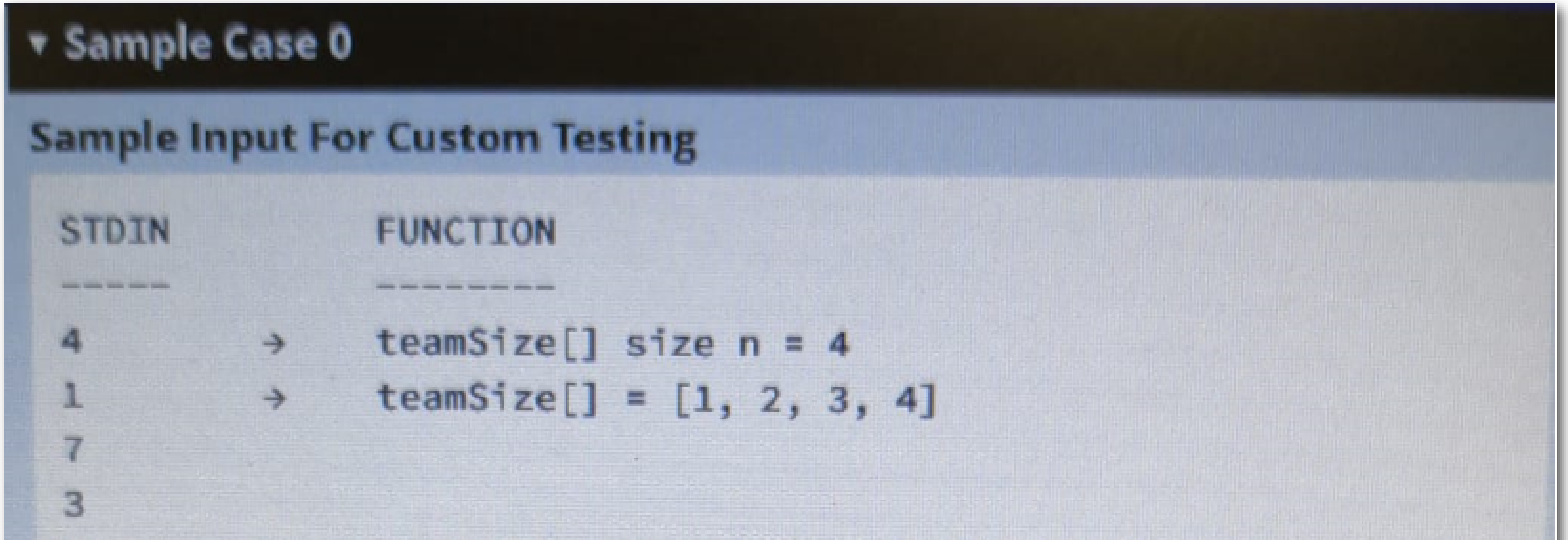
Constraints

- $1 \le n \le 2 * 10^5$
- $1 \le teamSize[i] \le 10^9$
- $0 \le k \le 10^9$

▾ Input Format For Custom Testing

The first line contains an integer, *n*, denoting the number of elements in *teamSize.*

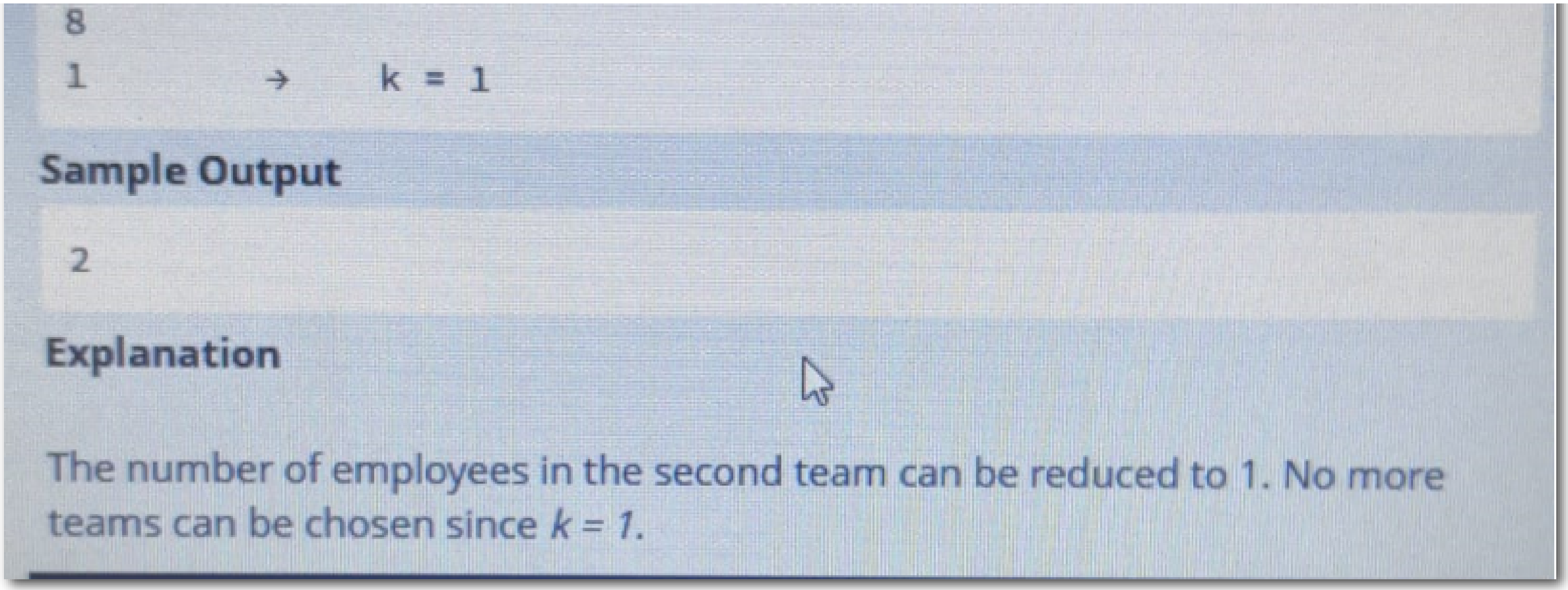
Each line *i* of the *n* subsequent lines (where $0 \le i < n$) contains an integer describing *teamSize[i].*
The next line contains an integer, *k*, denoting the maximum number of teams possible which have equal size.



▾ Sample Case 0

Sample Input For Custom Testing

```
STDIN          FUNCTION
-----          --------
4        →     teamSize[] size n = 4
1        →     teamSize[] = [1, 2, 3, 4]
7
3
```



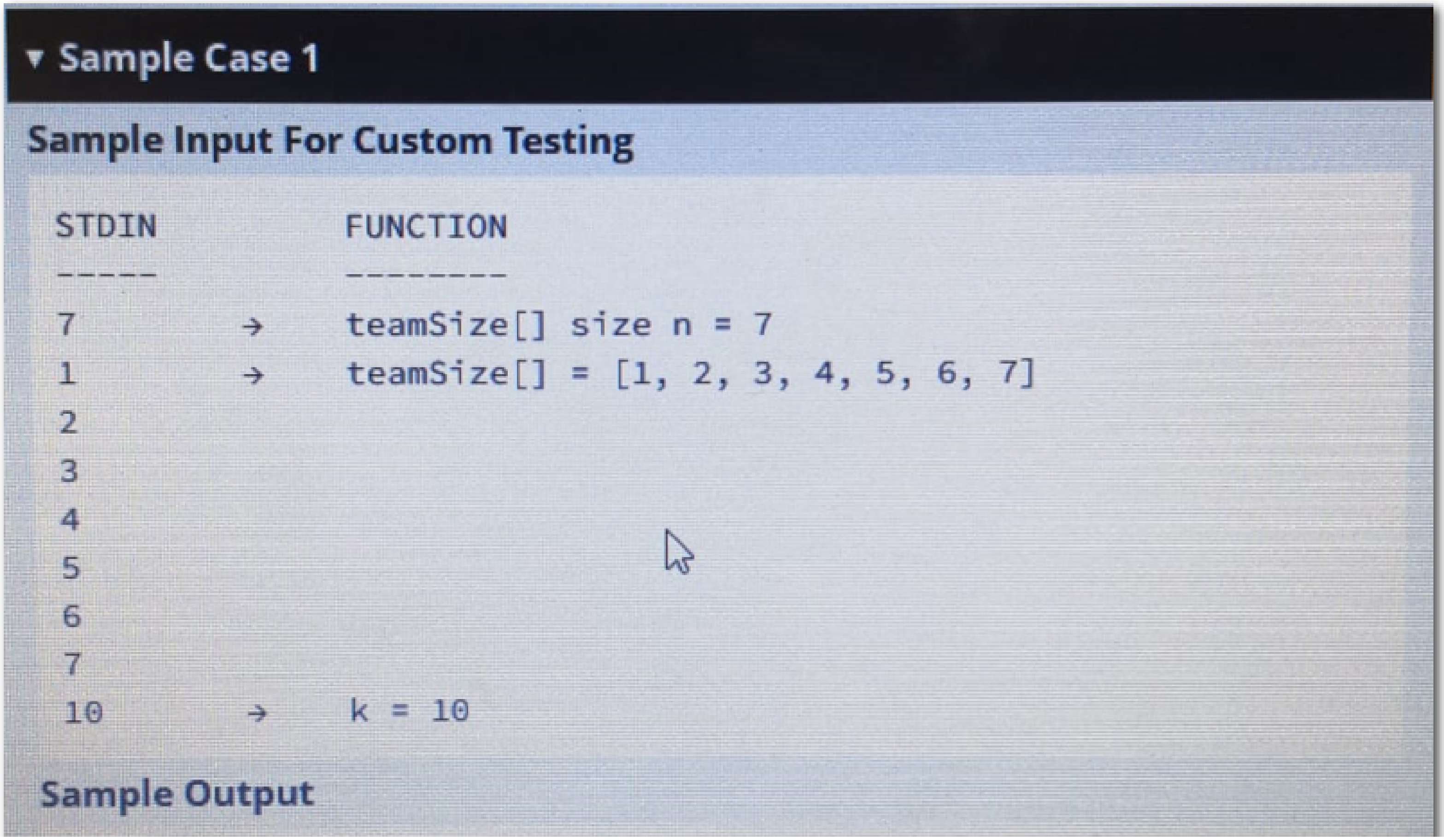
```
8
1           →     k = 1
```

Sample Output

```
2
```

Explanation

The number of employees in the second team can be reduced to 1. No more teams can be chosen since *k = 1.*

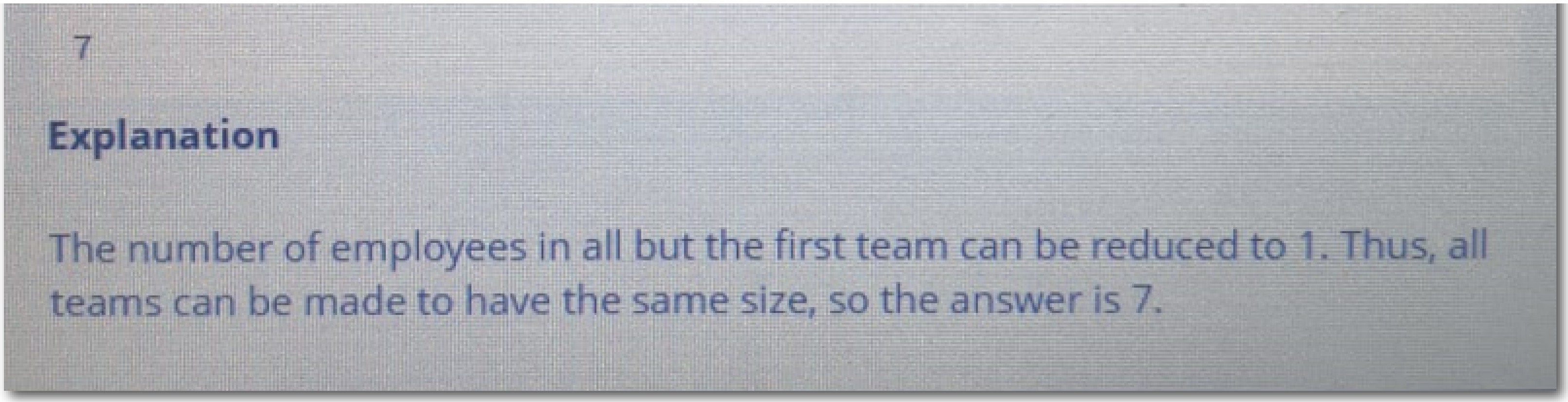


▾ Sample Case 1

Sample Input For Custom Testing

```
STDIN           FUNCTION
-----           --------
7          →    teamSize[] size n = 7
1          →    teamSize[] = [1, 2, 3, 4, 5, 6, 7]
2
3
4
5
6
7
10         →    k = 10
```

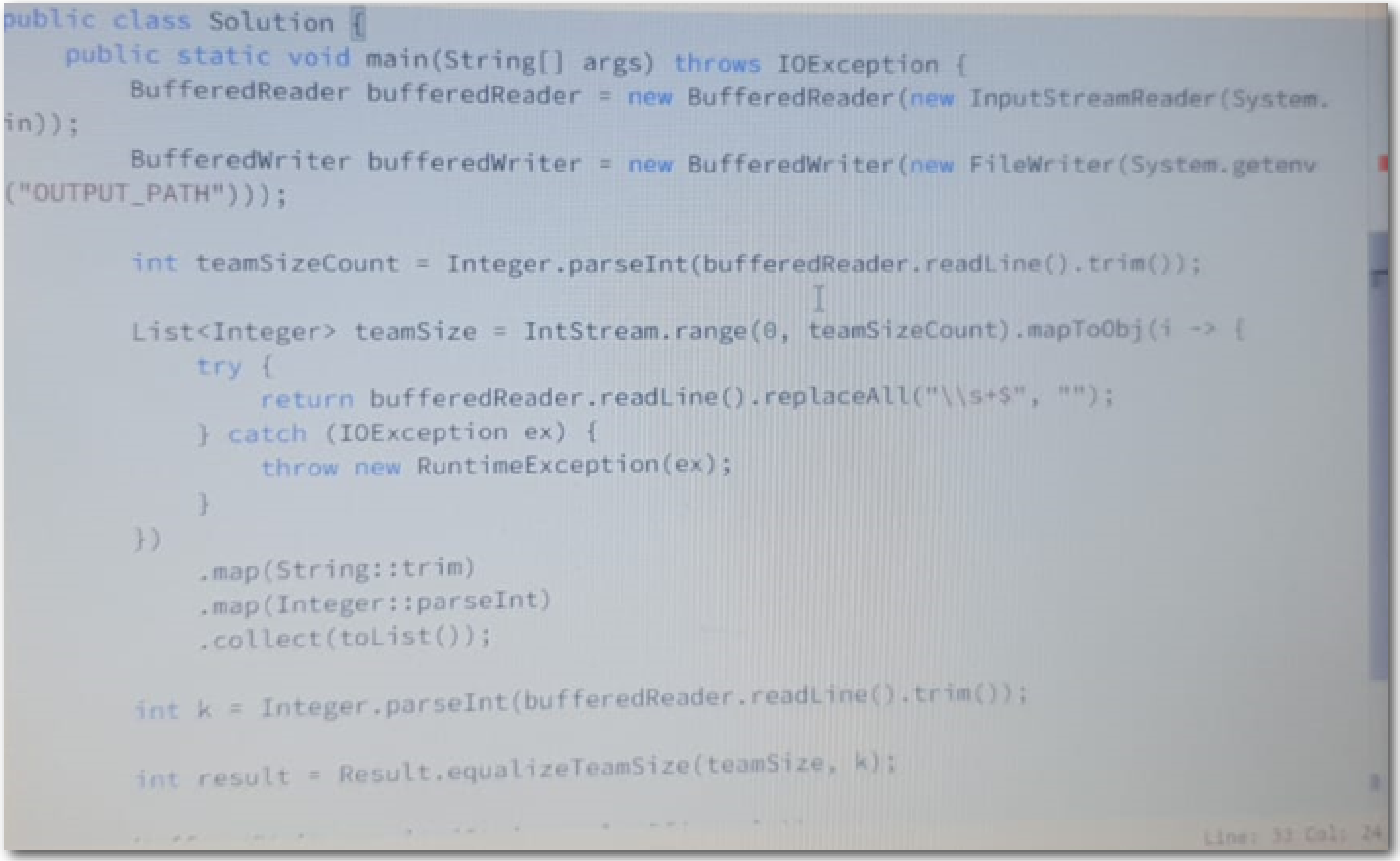
Sample Output



7

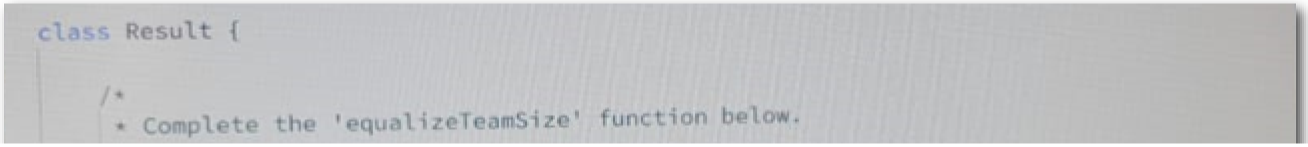
Explanation

The number of employees in all but the first team can be reduced to 1. Thus, all teams can be made to have the same size, so the answer is 7.

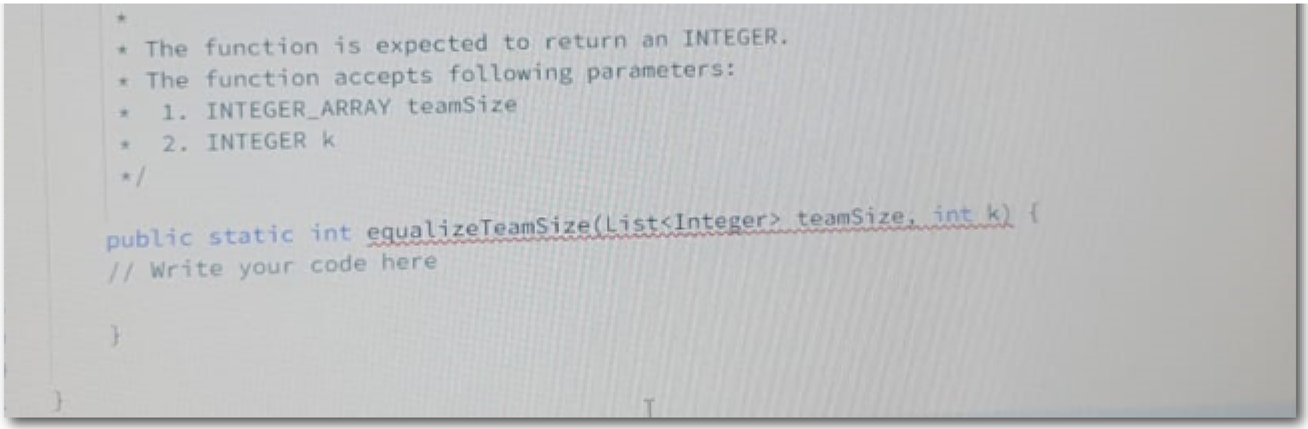


```
public class Solution {
    public static void main(String[] args) throws IOException {
        BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(System.
in));
        BufferedWriter bufferedWriter = new BufferedWriter(new FileWriter(System.getenv
("OUTPUT_PATH")));

        int teamSizeCount = Integer.parseInt(bufferedReader.readLine().trim());

        List<Integer> teamSize = IntStream.range(0, teamSizeCount).mapToObj(i -> {
            try {
                return bufferedReader.readLine().replaceAll("\\s+$", "");
            } catch (IOException ex) {
                throw new RuntimeException(ex);
            }
        })
            .map(String::trim)
            .map(Integer::parseInt)
            .collect(toList());

        int k = Integer.parseInt(bufferedReader.readLine().trim());

        int result = Result.equalizeTeamSize(teamSize, k);
```

Line: 33 Col: 24



```
class Result {

    /*
     * Complete the 'equalizeTeamSize' function below.
```



```
     *
     * The function is expected to return an INTEGER.
     * The function accepts following parameters:
     *  1. INTEGER_ARRAY teamSize
     *  2. INTEGER k
     */

    public static int equalizeTeamSize(List<Integer> teamSize, int k) {
    // Write your code here

    }

}
```