# Mark and Toys

Your Mark and Toys submission got 35.00 points. Share Tweet

Try the next challenge

**Problem** | Submissions | Leaderboard | Editorial | Topics

Mark and Jane are very happy after having their first child. Their son loves toys, so Mark wants to buy some. There are a number of different toys lying in front of him, tagged with their prices. Mark has only a certain amount to spend, and he wants to maximize the number of toys he buys with this money.

Given a list of prices and an amount to spend, what is the maximum number of toys Mark can buy? For example, if $prices = [1, 2, 3, 4]$ and Mark has $k = 7$ to spend, he can buy items $[1, 2, 3]$ for $6$, or $[3, 4]$ for $7$ units of currency. He would choose the first group of $3$ items.

**Function Description**

Complete the function maximumToys in the editor below. It should return an integer representing the maximum number of toys Mark can purchase.

maximumToys has the following parameter(s):

- prices: an array of integers representing toy prices
- k: an integer, Mark's budget

**Input Format**

The first line contains two integers, $n$ and $k$, the number of priced toys and the amount Mark has to spend.
The next line contains $n$ space-separated integers $prices[i]$

**Constraints**

$1 \leq n \leq 10^5$
$1 \leq k \leq 10^9$
$1 \leq prices[i] \leq 10^9$
A toy can't be bought multiple times.

**Output Format**

An integer that denotes the maximum number of toys Mark can buy for his son.

**Sample Input**

```
7 50
1 12 5 111 200 1000 10
```

**Sample Output**

```
4
```

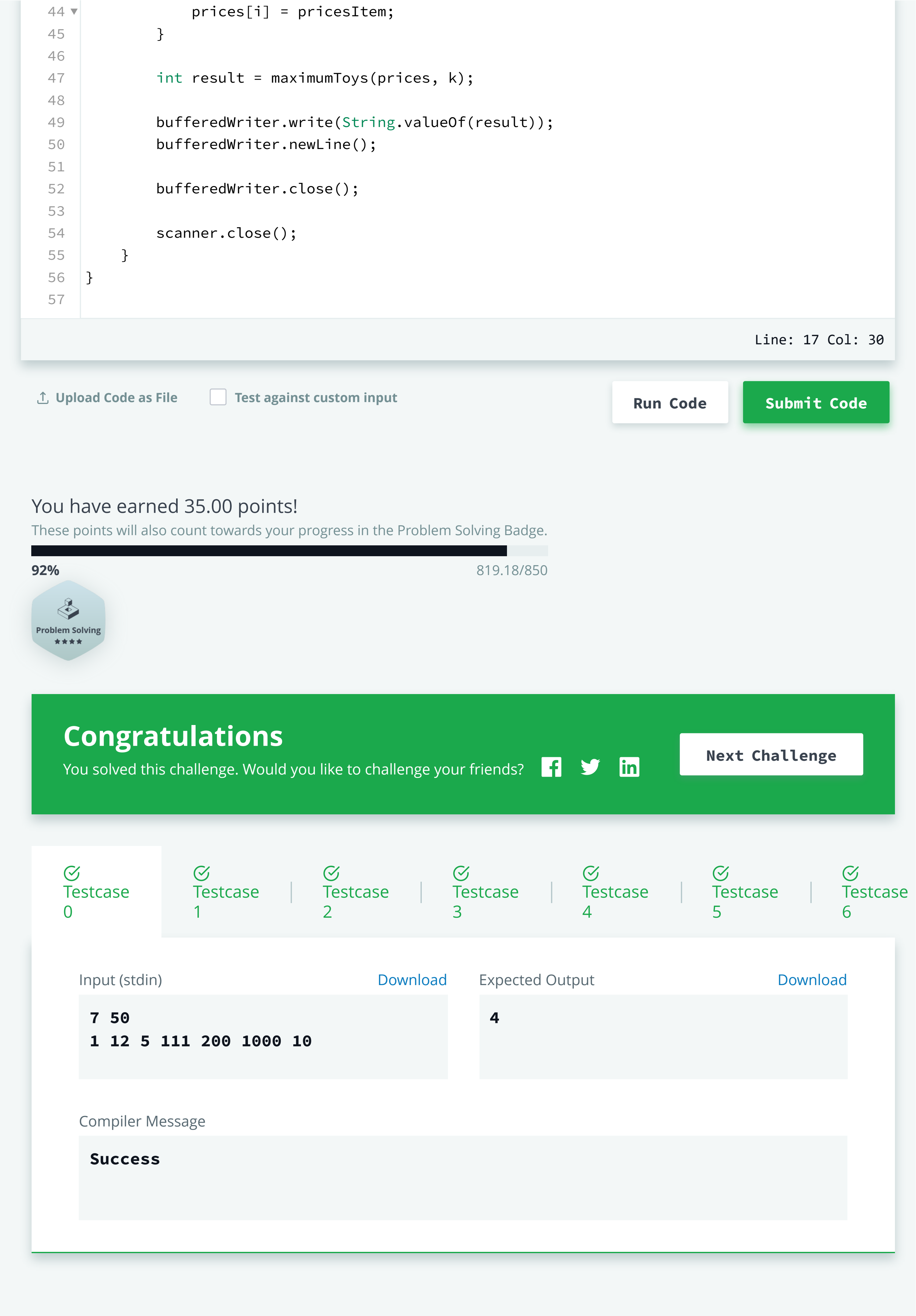
**Explanation**

He can buy only $4$ toys at most. These toys have the following prices: $1, 12, 5, 10$.

**Current Buffer** (saved locally, editable)

Java 8

```java
import java.io.*;
import java.math.*;
import java.security.*;
import java.text.*;
import java.util.*;
import java.util.concurrent.*;
import java.util.regex.*;

public class Solution {

    // Complete the maximumToys function below.
    static int maximumToys(int[] prices, int k) {
        Arrays.sort(prices);
        int count = 0;
        for(int i=0; i<prices.length; i++) {
            if(k-prices[i] >= 0) {
                k-=prices[i];
                count++;
            }else{
                return count;
            }
        }
        return count;
    }

    private static final Scanner scanner = new Scanner(System.in);

    public static void main(String[] args) throws IOException {
        BufferedWriter bufferedWriter = new BufferedWriter(new FileWriter(System.getenv("OUTPUT_PATH")));

        String[] nk = scanner.nextLine().split(" ");

        int n = Integer.parseInt(nk[0]);

        int k = Integer.parseInt(nk[1]);

        int[] prices = new int[n];

        String[] pricesItems = scanner.nextLine().split(" ");
        scanner.skip("(\r\n|[\n\r\u2028\u2029\u0085])?");

        for (int i = 0; i < n; i++) {
            int pricesItem = Integer.parseInt(pricesItems[i]);
```



```
            prices[i] = pricesItem;
        }

        int result = maximumToys(prices, k);

        bufferedWriter.write(String.valueOf(result));
        bufferedWriter.newLine();

        bufferedWriter.close();

        scanner.close();
    }
}

```

Line: 17 Col: 30

You have earned 35.00 points!

These points will also count towards your progress in the Problem Solving Badge.

92%

819.18/850

## Congratulations

You solved this challenge. Would you like to challenge your friends?

Input (stdin)

```
7 50
1 12 5 111 200 1000 10
```

Expected Output

```
4
```

Compiler Message

```
Success
```