



Stone Division, Revisited

by [nabila_ahmed](#)

Problem

Submissions

Leaderboard

Discussions

Editorial

You have a pile of n stones that you want to split into multiple piles, as well as a set, S , of m distinct integers. We define a *move* as follows:

- First, choose a pile of stones. Let's say that the chosen pile contains y stones.
- Next, look for some $x \in S$ such that $x \neq y$ and y is divisible by x (i.e., x is a factor of y); if such an x exists, you can split the pile into $\frac{y}{x}$ equal smaller piles.

You are given q queries where each query consists of n and S . For each query, calculate the maximum possible number of moves you can perform and print it on a new line.

Input Format

The first line contains an integer, q , denoting the number of queries. The $2 \cdot q$ subsequent lines describe each query in the following format:

- The first line contains two space-separated integers describing the respective values of n (the size of the initial pile in the query) and m (the size of the set in the query).
- The second line contains m distinct space-separated integers describing the values in set S .

Constraints

- $1 \leq q \leq 10$
- $1 \leq n \leq 10^{12}$
- $1 \leq m \leq 1000$
- $1 \leq s_i \leq 10^{12}$

Subtask

- $1 \leq m \leq 10$ for 30% of the maximum score.

Output Format

For each query, calculate the maximum possible number of moves you can perform and print it on a new line.

Sample Input 0

```
1
12 3
2 3 4
```

Sample Output 0

```
4
```

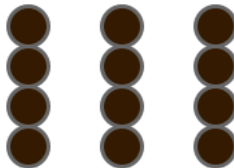
Explanation 0

Initially there is a pile with **12** stones:

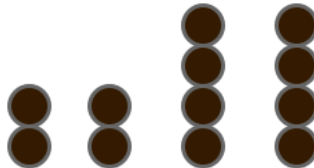


You can make a maximal **4** moves, described below:

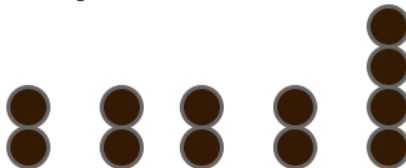
- Select $x = 4$ from S and split it into $\frac{12}{4} = 3$ equal piles of size **4** to get:



- Select $x = 2$ from S and split a pile of size **4** into $\frac{4}{2} = 2$ equal piles of size **2** to get:



- Repeat the previous move again on another pile of size **4** to get:



- Repeat the move again on the last pile of size **4** to get:



As there are no more available moves, we print **4** (the number of moves) on a new line.

f t in

Submissions: [544](#)

Max Score: 50

Difficulty: Medium

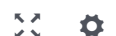
Rate This Challenge:



[More](#)

Current Buffer (saved locally, editable)

Java 7



```
1 import java.io.*;
```

```
2 import java.util.*;
3 import java.text.*;
4 import java.math.*;
5 import java.util.regex.*;
6
7 public class Solution {
8
9     static long maxPossibleMoves(long n, long[] s) {
10         // Complete this function
11     }
12
13     public static void main(String[] args) {
14         Scanner in = new Scanner(System.in);
15         int q = in.nextInt();
16         for(int a0 = 0; a0 < q; a0++){
17             long n = in.nextLong();
18             int m = in.nextInt();
19             long[] s = new long[n];
20             for(int s_i = 0; s_i < n; s_i++){
21                 s[s_i] = in.nextLong();
22             }
23             long result = maxPossibleMoves(n, s);
24             System.out.println(result);
25         }
26         in.close();
27     }
28 }
29
```

Line: 1 Col: 1

 [Upload Code as File](#) ☐ Test against custom input

[Run Code](#)[Submit Code](#)

Join us on IRC at [#hackerrank](#) on freenode for hugs or bugs.

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)