

Novice Problem

# Stamps

14 March 2009

---

Source File	<code>stamps.{java,c,cc}</code>
Input File	<code>stamps.in</code>
Output File	<i>standard output</i>

---

Everybody hates Raymond. He's the largest stamp collector on planet Earth; and, because of that, he always makes fun of all the others at the stamp collector parties. Fortunately, everybody loves Lucy; and, she has a plan. She secretly asks her friends whether they could lend her some stamps, so that she can embarrass Raymond by showing an even larger collection than his.

Raymond is so sure about his superiority that he always tells how many stamps he'll show. And since Lucy knows how many she owns, she knows how many more she needs. She also knows how many friends would lend her some stamps and how many each would lend. But she'd like to borrow from as few friends as possible and if she needs too many then she'd rather not do it at all. Can you tell her the minimum number of friends she needs to borrow from?

## Input

The first line contains the number of scenarios. Each scenario describes one collectors' party with two lines of input. The first line contains two integers, separated by a space, indicating how many stamps (from 1 to 1000000) Lucy needs to borrow and how many friends (from 1 to 1000) offer her some stamps. The second line contains space-separated integers, one for each lending friend, telling you how many stamps (from 1 to 10000) the friend will loan Lucy.

## Output

The output for every scenario begins with a line containing

Scenario  $N$  :

where  $N$  is the number of the scenario starting with 1. Then print a space and the minimum number of friends Lucy needs to borrow stamps from. If it's impossible even if she borrows everything from everybody, write

impossible

instead of  $N$ .

Output is emitted to standard output, with no leading or trailing spaces.

C, C++	<code>stdout</code>
C++	<code>cout</code>
Java	<code>System.out</code>

## Example

Sample input and output are given in figures 1 and 2, respectively.

```
3
100_6
13_17_42_9_23_57
99_6
13_17_42_9_23_57
1000_3
314_159_265
```

Figure 1: Input

```
Scenario_#1:_3
Scenario_#2:_2
Scenario_#3:_impossible
```

Figure 2: Output