

2135 Sebastian's Problem

Once upon a time there lived a programmer with the potential to be the best Java programmer the world had ever seen, and his name was Sebastian. Unfortunately for Sebastian, he was born several decades prematurely and was forced to store all of his programs on punchcards. If you do not know what a punchcard is ask one of your professors after the contest. We are sure she will be able to tell you many horror stories about punchcards and shoe boxes.

One day Sebastian had to turn in his senior project, a program that calculated π to a billion decimal places. On his way to deliver his program to the grader, he was suddenly overtaken by a vicious case of scurvy (pizza just doesn't have that much vitamin C). In his mad dash to the nearest lavatory, he dropped the box of punchcards containing his program, and the cards scattered everywhere.

After he had picked up all the cards, and counted them, he realized that one was missing. Luckily Sebastian had taken the time to record his program card-by-card in a notebook at the dorm. This will allow him to replicate the missing card and still turn his project in on time. The problem was that the cards were all jumbled and he didn't know which card to replicate.

Here are a few things that Sebastian knows about his cards:

- Sebastian's program had at least 1 card before he dropped it.
- There are n cards numbered from 1 to n .
- Exactly one card is missing.
- No cards have duplicate numbers.

Please write a program to help Sebastian determine what card number his needs to replicate so that he can turn-in his project on time.

Input

Input will consist of up to 100 data sets. Each data set will consist of:

1. *Start line* — A single line of the form 'START n ' where n is the number of punch cards in the program. No program will consist of more than 10,000 cards.
2. *Card list* — A list of $n - 1$ card numbers in the range 1 to n (inclusive). Each card number will appear on its own line with no leading or following whitespace.
3. *End line* — A single line of the form 'END'.

Output

For each data set, output will consist of a single number on its own line representing the card that was missing from the *Card list* of that data set.

Note: these samples are **VERY** small compared to some of the data sets your problem will be judged against and are abbreviated due to the number of trees that would have to be destroyed to provide a large example. The limit is 10,000 cards per data set.

Sample Input

```
START 1
END
START 5
1
2
4
5
END
START 20
13
14
17
1
2
3
4
15
16
5
18
20
6
10
11
12
7
8
9
END
```

Sample Output

```
1
3
19
```