ToLower

Problem ID: tolower
CPU Time limit: 1 secon
Memory limit: 1024 MB
Difficulty: 1.8

Author: Tony Cai
Source: Calgary Collegiat Programming Contest 20

Anthony is participating in a programming contest today! He’s excellent at algorithms; he can design and implement the solution to even the hardest problems in the programming contest in minutes. Unfortunately, parsing inputs is his greatest weakness. Specifically, he really struggles with problems which require him to convert strings to lowercase. When Anthony is given a string, he only converts the first character of the string to lowercase.

You somehow gained access to the test data for every problem in today’s contest. There are P problems in the contest, and every problem in the contest has exactly T test cases. Each test case consists of a single string with only English characters, and every problem requires converting the entire string to lowercase as a preprocessing step.

As Anthony is a master problem solver, you know that he is able to design and implement the algorithm for every problem within the contest’s time constraint. Specifically, Anthony’s program will pass a test case C if and only if every character in the string corresponding to C is converted to lowercase. Now you wonder: how many problems will he solve in the contest? Note that a problem is considered solved if all test cases passed.

Input

The first line of the input contains two integers P and T, $1 \leq P, T \leq 50$.

The next PT lines contain the test data for all the problems in the contest.

Every line contains exactly one nonempty string composed of only English characters. Test cases are grouped by problems, so the first T lines represent all the test cases for problem 1, the next T lines represent all the test cases for problem 2, and so on. It is guaranteed that the total number of characters in the test cases does not exceed $50\,000$.

Output

Output a single integer denoting the number of problems Anthony will solve during the contest.

Sample Input 1

```
2 2
abc
Def
DDG
add
```

Sample Output 1

```
1
```