Hot Air Ballooning

`balloon.c`, `balloon.cpp`, `balloon.c11`, `Balloon.java`, `balloon.py`

Dave is the director of the Summer school of hot air ballooning. Being a responsible director, he keeps a list of flights of each trainee in the school. After each flight, Dave appends a note to the lists of flights of each trainee participating in that particular flight. The note is very simple, it just indicates the type number of the balloon. In this way, each trainee flight history is characterized by a list of numbers.

At the end of the season, Dave wants to categorize the trainees according to their experience with different brands of balloons.

Two trainees belong to the same category if they have flown the same types of balloons. It does not matter how many times they have flown any particular balloon type, what does matter is the set of the balloon types they have flown and that has to be the same.

There are exactly nine types of balloons in Dave's school, and no trainee has flown more than nine times in a balloon, so Dave expresses each trainee list as an integer consisting of digits $1, 2, \ldots, 9$ and smaller than one billion. He thinks that this representation will help him to process the lists programmatically by a computer.

For example, the trainees characterized by integers 234423 and 342 belong to the same category, while the trainees characterized by integers 118821 and 1189821 belong to different categories.

Help Dave to calculate how many different categories of trainees attended the school this season.

Input Specification

There are more test cases. Each case starts with a line containing one integer N ($1 \leq N \leq 1\,000$) representing the number of trainees. Next, there are N lines, each line contains one integer representing the list of flights of one particular trainee.

Output Specification

For each test case, print a single line with one integer C specifying the number of different trainee categories in the school.

Sample Input

```
5
132
42
3312
43
24424
3
222
22
2
```

Output for Sample Input

```
3
1
```