Magic Spells

While playing a video game, you are battling a powerful dark wizard. He casts his spells from a distance, giving you only a few seconds to react and conjure your counterspells. For a counterspell to be effective, you must first identify what kind of spell you are dealing with.

The wizard uses scrolls to conjure his spells, and sometimes he uses some of his generic spells that restore his stamina. In that case, you will be able to extract the name of the scroll from the spell. Then you need to find out how similar this new spell is to the spell formulas written in your spell journal.

Spend some time reviewing the locked code in your editor, and complete the body of the *counterspell* function.

Check Dynamic cast to get an idea of how to solve this challenge.

Input Format

The wizard will read t scrolls, which are hidden from you.
Every time he casts a spell, it's passed as an argument to your *counterspell* function.

Constraints

- $1 \leq t \leq 100$
- $1 \leq |s| \leq 1000$, where s is a scroll name.
- Each scroll name, s, consists of uppercase and lowercase letters.

Output Format

After identifying the given spell, print its name and power.
If it is a generic spell, find a subsequence of letters that are contained in both the spell name and your spell journal. Among all such subsequences, find and print the length of the longest one on a new line.

Sample Input

```
3
fire 5
AquaVitae 999 AruTaVae
frost 7
```

Sample Output

```
Fireball: 5
6
Frostbite: 7
```

Explanation

Fireball and *Frostbite* are common spell types.
AquaVitae is not, and when you compare it with *AruTaVae* in your spell journal, you get a sequence: