Problem H: Word ladder

You now work for a puzzle company. They have a puzzle they call a *Word Ladder*. A solver starts with a given starting word, and makes changes one letter at a time until s/he reaches a target word, with no word in the chain appearing more than once. There are three ways to take a single step from one word to another:

- Change one letter
- Add one letter
- Remove one letter

So, it's one step from COT to CAT, one step from CAT to SCAT, and one step from SCAT to SAT. Here's a word ladder from COT to SCAT:

COT ⇒ CAT ⇒ SAT ⇒ SCAT

Here's another word ladder from COT to SCAT:

COT ⇒ CAT ⇒ SCAT

The length of a word ladder is the number of words in it, so the examples above show a word ladder of length 4, and one of length 3. The second is the shortest possible between COT and SCAT. Shorter ladders are considered better than longer ladders.

The puzzle company knows that, given two words, a smart solver will always find the best ladder, which is the shortest ladder, between them. They want to give their solvers a challenge, so they are looking for long word ladders. Given a limited vocabulary, you need to tell them the length of the longest word ladder that a smart solver would find using only words in that vocabulary - that is, the longest of all best ladders.

Input

Input will consist of multiple datasets. Each dataset starts with an integer N ($1 \leq N \leq 500$) which indicates the number of words in the vocabulary. The words follow, one per line.

Each word will consist only of 1 to 50 lower-case letters. There will be no other characters or white space.

The end of input is indicated by a line containing a single zero.

Output

For each input set, print a line containing a single integer representing the length of the longest ladder that a smart solver would find.

Example

Given the input

```
4
cat
cot
scat
sat
7
welcome
to
the
acm
regional
programming
contest
0
```

the output would be

```
3
1
```