AB-words

Every sequence of small letters a and b (also the empty sequence) is called an ab-word. If $X = [x_1, ..., x_n]$ is an ab-word and i, j are integers such that $1 <= i <= j <= n$ then $X[i..j]$ denotes the subword of X consisting of the letters $x_i, ..., x_j$. We say that an ab-word $X = [x_1..x_n]$ is nice if it has as many letters a as b and for all $i = 1, ..., n$ the subword $X[1..i]$ has at least as many letters a as b.

Now, we give the inductive definition of the similarity between nice ab-words.

- Every two empty ab-words (i.e. words with no letters) are similar
- Two non-empty nice ab-words $X = [x_1, ..., x_n]$ and $Y = [y_1, ..., y_m]$ are similar if they have the same length ($n = m$) and one of the following conditions if fulfilled:
  1. $x_1 = y_1$, $x_n = y_n$ and $X[2..n\text{-}1]$ and $Y[2..n\text{-}1]$ are similar ab-words and they are both nice;
  2. there exists i, $1 <= i <= n$, such that $X[1..i]$, $X[i+1..n]$ are nice ab-words and
     a. $Y[1..i]$, $Y[i+1..n]$ are nice ab-words and $X[1..i]$ is similar to $Y[1..i]$ and $X[i+1..n]$ is similar to $Y[i+1..n]$, or
     b. $Y[1..n\text{-i}]$, $Y[n\text{-}i+1..n]$ are nice ab-words and $X[1..i]$ is similar to $Y[n\text{-}i+1..n]$ and $X[i+1..n]$ is similar to $Y[1..n\text{-}i]$.

A **level of diversity** of a non-empty set S of nice ab-words is the maximal number of ab-words that can be chosen from S in such a way that for each pair w_1,w_2 of chosen words, w_1 is not similar to w_2.

Task

Write a program that for each test case:

- reads elements of S from standard input;
- computes the level of diversity of the set S;
- writes the result to standard output.

Input

The number of test cases t is in the first line of input, then t test cases follow separated by an empty line.

In the first line of a test case there is a number n of elements of the set S, $1 <= n <= 1000$; in the following n lines there are elements of the set S, i.e. nice ab-words (one word in each line); the first letter of every ab-word is the first symbol in line and there are no spaces between two consecutive letters in the word; the length of every ab-word is an integer from the range [1..200].

Output

For each test case your program should output one line with one integer - the level of diversity of S.

Example

Sample input:

```
1
3
aabaabbbab
abababaabb
abaaabbabb
```

Sample output:

```
2
```