Problem D
The Big Sorter

Source file: `sorter.{c | cpp | java}`
Input file: `sorter.in`

The Borrows-Wheeler transform is a technique used in compression. The first step of the transform requires the sorting all the possible rotations of the input data. If this were to be applied to the string “CONTEST”, we would get the following rotations:

CONTEST
ONTESTC
NTESTCO
TESTCON
ESTCONT
STCONTE
TCONTES

Which when sorted would produce the sequence:

CONTEST
ESTCONT
NTESTCO
ONTESTC
STCONTE
TCONTES
TESTCON

with the original string at position 0.

Your task is to calculate the position of the original string in the sorted sequence of the rotations. If a rotation matches the original string, the original is considered smaller.

Input

The first line of the input file contains an integer N ($0 < N < 100$) indicating the number of test cases. Each test case consists of one line containing an arbitrary string of ASCII characters. The maximum line length is 64kB.

Output

For each test case you should output in a separate line, the position of the original string in the sorted sequence of all the possible rotations.

Sample Input

```
2
CONTEST
PROGRAMMING
```

Output for Sample Input

```
0
8
```