

Jumping Yudi

Yudi is invited to a party by his crush. Yudi is a simple man. Yudi must come.

To go to the party from his home, Yudi has to cross a great river; the alternative is just too far away. Fortunately, there are some footholds across the river where one can stand above those.

Did I mention that Yudi is an talented engineer? He created a jumping machine which able to (of course) jump with a fix distance. For example, consider the following river layout of 1..10 ('*' means foothold, '.' means water). Yudi always starts at the riverside, at point 0, and his goal is to reach any land greater than point 10 (the river spans from 1 to 10). Need not to say that his machine is not waterproof, so it should not jump into the water.

```
Y * . * * * * . . . .
0 1 2 3 4 5 6 7 8 9 10
```

If he configures his machine to jump at distance of 3, then he will surely fall at point 9.

```
0-----1-----2-----x
Y * . * * * * . . | .
0 1 2 3 4 5 6 7 8 9 10
```

Similar things happen when he configure his jumping machine to jump at distance of 1 (fall at 2), 2 (fall directly), 4 (fall at 8), 5 (fall at 10), 7, 8, 9, and 10.

One possible configuration is by setting his machine to jump at distance of 6. Thus, he will safely land at position 12 (greater than 10) on the other side. The other possibility is setting his machine to jump at distance of greater than 10, reaching the other side straight away.

```
0-----1-----
Y * . * * * * . . . .
0 1 2 3 4 5 6 7 8 9 10
```

This jumping machine consumes a lot of energy and it is linear to its jumping distance (not to how many times does it jump). Energy is expensive, we should conserve. Help Yudi determine the smallest jumping distance he has to set to his jumping machine in order to reach the other side of the river (of 1..N) from point 0.

Input

Input begins with an integer T ($1 \leq T \leq 20$) in a line denoting the number of cases. Each case contains a string S in a single line denoting the river's layout. S consists of only character '*' (foothold) and '.' (water), and its length is between 1 and 1,000,000, inclusive.

Output

For each test case, output "Case #X: Y" in a line, where X is the case number (starts from 1), and Y is the output for the respective case.

Sample input

```
5
* .****. ....
.....
*****
* .****. .*. .
.*.*.*.*.*..
```

Output for sample input

```
Case #1: 6
Case #2: 6
Case #3: 1
Case #4: 3
Case #5: 4
```