Code Jam to I/O for Women 2020 - Code Jam to I/O for Women

Interleaved Output: Part 2

Problem

*You do not need to read the **Interleaved Output: Part 1** problem to be able to solve this problem. Both Part 1 and Part 2 have the same first two paragraphs (not including this informational text). We have underlined the critical difference between the two parts.*

On a distant moon of Jupiter, some developer conference events are about to happen! They are called `IO` (uppercase I, uppercase O), `Io` (uppercase I, lowercase o), `iO` (lowercase I, uppercase O), and `io` (lowercase I, lowercase O).

The best way to advertise an event is by using special computers that print the event's name one character at a time, with the output appearing on a digital display. Each such computer only knows the name of one event, and is programmed to print its event's name zero or more times. For example, a computer programmed to print `IO` twice prints an `I`, followed by an `O`, followed by an `I`, followed by an `O`, for a final string of `IOIO`.

You know that the conference organizers are using exactly one computer to advertise each event. Each printer may print its event name zero or more times. Moreover, the computers are not necessarily all programmed to print the same number of times.

The computers have all finished printing, but unfortunately, they all printed to the same display! Because the computers printed concurrently, event names in the final output string may be interleaved. You are considering the possible ways in which that string could have been produced.

For example, the string `IiOioIoO` could have been produced as follows:

```
index:  1 2 3 4 5 6 7 8
IO:     . . . . . . . .
Io:     I . . . o I o .
iO:     . i O i . . . O
io:     . . . . . . . .
string: I i O i o I o O
```

In this interpretation, the `Io` event was advertised twice, the `iO` event was advertised twice, and the other two events were not advertised at all.

Notice that there is no valid interpretation of this string in which the `IO` computer advertised its event twice. In that case, the remaining output, `iioo`, would have had to have come from the `io` computer, but that is impossible — that computer would have had to have printed `i` twice in a row, which is not allowed.

However, it is possible that the `IO` computer advertised its event once, as in the following interpretation:

```
index:  1 2 3 4 5 6 7 8
IO:     . . . . . I . O
Io:     I . . . o . . .
iO:     . i O . . . . .
io:     . . . i . . o .
```

```
    string: I i O i o I o O
```

Given a final output string, what is the maximum possible number of times that the event `IO` could have been advertised?

It is guaranteed that the string has at least one valid interpretation. For example, `oI`, `IOI`, and `IIOO` are not valid inputs.

Input

The first line of the input gives the number of test cases, **T**. **T** lines follow; each represents a single test case. Each case consists of a string **S** containing only the characters from the set `I`, `O`, `i`, and `o`.

Output

For each test case, output one line containing `Case #x: y`, where `x` is the test case number (starting from 1) and `y` is the maximum number of times `IO` could have been advertised, as described above.

Limits

Time limit: 20 seconds per test set.
Memory limit: 1GB.
1 ≤ **T** ≤ 100.
The length of **S** is even.
There is at least one interpretation of the string that is consistent with the rules above. (That is, there is some way to assign each character to one of the four computers, such that the substring corresponding to each computer consists of zero or more repeats of that computer's event's name.)

Test set 1 (Visible Verdict)

2 ≤ the length of **S** ≤ 8.

Test set 2 (Hidden Verdict)

2 ≤ the length of **S** ≤ 100.

Sample

Input	Output

`5`
`IiOioIoO`
`IIOiOo`
`IoiOiO`
`io`
`IiOIOIoO`
	`Case #1: 1`
	`Case #2: 1`
	`Case #3: 0`
	`Case #4: 0`
	`Case #5: 3`

Sample Case #1 is the one described in the problem statement. (If you have read Interleaved Output: Part 1, notice that it is the same input as in the first sample case in that problem, but the output is different.)

In Sample Case #2, it is not possible that `IO` was advertised twice, because then the `IO` computer would have had to print two `I`s in a row. However, it is possible that `IO` was advertised once, e.g.:

```
index:  1 2 3 4 5 6
IO:     I . O . . .
Io:     . I . . . o
iO:     . . . i O .
io:     . . . . . .
string: I I O i O o
```

In Sample Case #3, notice that it is not possible that `IO` was advertised. The second character, `o`, must have been printed by the same computer that printed the first character `I`.

In Sample Case #4, notice that it is possible that `I` and/or `O` might not even show up in the string.

In Sample Case #5, it is possible that `IO` was advertised as many as three times (and in that case, `io` was advertised once).