Nowadays emoticon has become an art. People are no longer limited to simple ones like ':-)', ':-(', ':-P', etc. They use '>:O', '~_~', '=^_^=' and so on. Recently I came across '^_^' and it looks kind of cute to me. Given a string S consisting of only '_'s and '^'s, I was wondering what is the maximum number of disjoint subsequences of "^_^" (quote for beauty) in the string S.

For example, if S = "^^__^^" then the answer is 2. However, for S = "_^^" the answer is 0.

Input

Input starts with a positive integer T ($\leq 5,000$), denoting the number of test cases. Hence follows T test cases. Each case consists of a single string made of only '^' and '_'. The length of the strings would be at most 100,000 and the sum of lengths of the strings will be 2,100,000 at most.

Output

For each test case, print the case number followed by the answer.

Hint:

- $S[1...n]$ means S is a string of length n and it is 1-indexed.
- S_i means i'th character of S.
- A string $S[1...n]$ is a subsequence of another string $T[1...m]$, if we can find: $(t_1, t_2, \ldots, t_n)$ such that, $S[i] = T[t_i]$ for $1 \leq i \leq n$ and $1 \leq t_1 < t_2 < \ldots < t_n \leq m$. For example, 'abc' is a subsequence of 'aabbcc' but not of 'bca'.
- Two subsequences are disjoint if same character (position matters) is not used in both of the subsequences. For example, let S = 'abca'. 'ab' and 'ca' are two disjoint subsequences of S. However, if S = 'abc' then 'ab' and 'ac' are not disjoint subsequences. In both of these examples the subsequences are unique. However, for S = 'aabb' let's form two subsequences S_1S_3 and S_2S_4 (both are 'ab'), both of these are disjoint. But if we have chosen S_1S_3 and S_1S_4 then they would not be disjoint.

Sample Input

```
5
_^^_^^_
^__^__^
______
^^__^^
^_^^_^
```

Sample Output

```
Case 1: 1
Case 2: 1
Case 3: 0
Case 4: 2
Case 5: 2
```