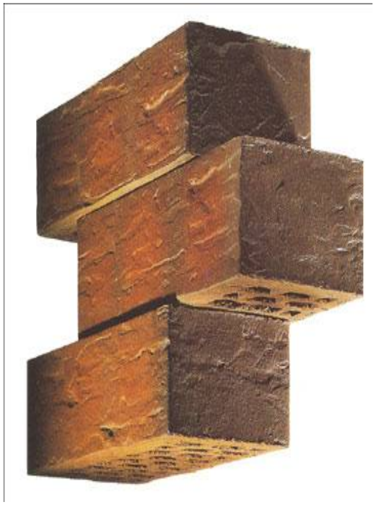
A

Brick Game

Input: Standard Input
Output: Standard Output

SH and **TR** are playing the famous *brick game*. For those who aren't aware of the rules of this illustrious game, here goes a brief description:



- It's a two player game in which there is initially some number of bricks on a table.
- Each player moves alternately. Player **1** makes the first move. In our case, **SH** is player **1**.
- In each move, a player removes some number of bricks from the table. The number of bricks that can be removed in a single move must be a member of a certain set of **K** numbers.
- The player to remove the last brick wins the game. If a player finds that he can't make any valid move in his turn, he loses the game.

Both **SH** and **TR** have recently spent a lot of time working out at the gym. They have acquired enough strength to lift a lot of bricks in a single move. However, the bricks that they are playing with are quite heavy and in a single move a player can remove at most 20 bricks from the table.

The set, thus, can be represented as **S** = {$0 < A_1 < A_2 < A_3 < \ldots < A_{K-1} < A_K < 21$}

You will be given a string consisting of **N** characters. The **i**th(**1 ≤ i ≤ N**) character represents the state if there were initially `i` bricks. A state is symbolized by **L** or **W** and they represent losing states and winning states, respectively. The states are calculated based on the fact that both of them play perfectly. For example, if the 10th character of the given string is **L**, then no matter what strategy **SH** follows he will always end up losing if they start with 10 bricks.

Given the states, your job is to minimize the cardinality of the set **S**; that is you have to minimize the value of **K**. Note that the set cannot be empty. You will also need to find the members of the set **S**. In case there are multiple sets with the same minimum cardinality that satisfies the given states, you have to select the one that comes lexicographically earliest. Set **S1** will come earlier than **S2,** if in the first differing element **S1**'s corresponding value has a lesser magnitude. Example: (1, 3, 6) will come before (1, 4, 5).

Input

The first line of input is an integer **T**(**T**<**201**) that indicates the number of test cases. Each of the next **T** lines contains a string consisting of characters **L** and **W** only. The length of a string is at most **100**.

Output

For each case, output the case number first. Then output the set of **K** elements in ascending order based on the description given above. It is guaranteed that there will be at least 1 set of numbers that will yield the given states in the input.

Sample Input

```
3
WWLWWL
WWWW
WLW
```

Output for Sample Input

```
Case 1: 1 2
Case 2: 1 2 3 4
Case 3: 1
```

Problemsetter: Sohel Hafiz
Special Thanks: Md. Arifuzzaman Arif