

String! String! String!

In computer science, String is one of the most important, interesting topics. String is used in Databases, compiler, Bio-Informatics etc. As a programmer and problem solver you should know about the string algorithms. There are several string algorithms and I am not going to discuss none of them. I am just telling you the names so that you can check those algorithms later.

1. Knuth–Morris–Pratt algorithm
2. Z algorithm
3. Rabin Karp Algorithm
4. Aho Corasick String Matching Algorithm
5. Manacher Algorithm
6. Boyer–Moore string search algorithm

And there are some data structures on string, they are –

1. Trie Tree
2. Suffix Array
3. Suffix Trie
4. Suffix Tree
5. Rope Data Structure

Now we need an algorithm which can be re-invented by you to solve the following problems. And that is-

You will be given several strings, for each string you need to find the closest string for that string. What is a closest string? A string is closest string of another string if prefix match between them is maximum and obviously not considering the string itself (BTW, this definition is mine, it can be different for different person J. But you have to follow this definition to solve this problem). Let's say you have four strings-

1. ABABABA
2. ABCDDAA
3. ABABKKK
4. BAAWEE

For string 1, the closest string is the string numbered 3.

For string 2, the closest string is the string numbered 1 and 3.

For string 3, the closest string is the string numbered 1

For string 4, there is no closest string.

Input

Input starts with an integer **T** (≤ 50), denoting the number of test cases.

Each case starts with a line containing an integer **N** ($1 \leq N \leq 100$) denoting the number of strings.

Each string will contain only the UPPERCASE English letters (A- Z). And each string will have length at most 100. And different strings may have different lengths. You can assume that there will be no spaces, symbols, punctuation marks in the input strings.

Output

For each case, print the case number in a single line. Then for each given string, print the index of the closest string in a single line. If there are multiple strings which are closest for a certain string print the minimum index. Indexes are starting from 1. And if there is no closest string then print -1. Check the output for the sample input for clear understanding.

Sample Input	Output for Sample Input
---------------------	--------------------------------

3	Case 1:
4	3
ABABABA	1
ABCDDAA	1
ABABKKK	-1
BAAWEEE	Case 2:
1	-1
A	Case 3:
3	2
A	1
A	1
A	

Note: Prefix is a substring of a string which is starting for the beginning. For ABCD,

1. A
2. AB
3. ABC
4. ABCD

are the prefixes. BCD is not a prefix of ABCD.