Dictionary Strings

Problem Statement

Gopal is preparing for a competitive exam and he has to prepare many topics for it. To remember the concepts better he identified a set of words from each topic. He prepared dictionaries for each of these topics with the set of identified words so that he can refer to them easily.

While recollecting the topics Gopal sometimes could not remember to which dictionary a certain word belongs. After all the hard work, Gopal didn't want to lose marks due to this confusion. So he requested his friend, Govind, to help him identify a way to check if a word belongs to a dictionary.

Govind, being a very good friend of Gopal, wants to help him do better in the exam. So, after some thought, he finally came up with a solution.

For each dictionary, a string is chosen from which all the words can be made by selecting a subset of the characters from the string and rearranging them. (It is not necessary that the characters are consecutive and/or in the same order as in the string). They called this string a `Dictionary String`. When confused about to which dictionary a word belongs, Gopal can check if the word can be extracted from the `Dictionary String` for that dictionary.

To qualify as a `Dictionary String`, all the letters needed to explicitly form each word of the dictionary must be present in the string. You cannot reuse letters. Thus, the string `aab` is not a Dictionary String for a dictionary containing the word `aaa` since this word needs 3 `a`'s whereas the candidate Dictionary String has only two `a`'s.

To help Gopal memorize the Dictionary Strings better, Govind inserted extra characters in some of the Dictionary Strings that appeared harder to memorize. To distinguish those strings from others he calls a string without any extra characters, a `Perfect Dictionary String`.

Govind would like your help in verifying his program. For a set of words in a dictionary, you should indicate whether a string is a perfect dictionary string and/or a dictionary string. If a word is not a dictionary string, he would like you to tell him the minimum number of characters needed to convert the string to a dictionary string.

Notes:

Some of the test cases are very large, and may require you to speed up input handling in some languages.

In C++, for example, you can include the following line as the first line in your main function to speed up the reading from input:

```
std::ios_base::sync_with_stdio (false);
```

And in Java, you can use a BufferedReader to greatly speed up reading from input, e.g.:

```
BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
// Read next line of input which contains an integer:
int T = Integer.valueOf(reader.readLine());
```

Input Format

Input begins with a single integer T, $1 <= T <= 100$, which denotes number of test cases.

Each test case begins with a line, which contains 2 space-separated integers D and S. D represents the number of words in a dictionary, and S represents the number of potential dictionary strings to be checked. Note that $1 <= D, S <= 100$.

Next follows D lines, each containing a word in the dictionary.

The remaining S lines in the test case each contain a potential dictionary string.

Notes: The words in the dictionary and the potential dictionary strings will consist of only lower-case letters. The lengths of these strings are greater than or equal to one character and less than or equal to 40,000 characters.

Output Format

For each of the S potential dictionary strings, you should output a line with two values separated by a space in the following format:

A_1 A_2

Where

- A_1 is either `Yes` or `No` denoting if a string is a Dictionary String or not.
- If A_1 is `No`, then A_2 is the minimum number of characters needed to make the string a Dictionary String. If A_1 is `Yes`, then A_2 is `Yes` if the string is a Perfect Dictionary String, and `No` otherwise.

Sample Input

```
1
5 3
ant
top
open
apple
lean
anteplop
antelope
penleantopan
```

Sample Output

```
Yes Yes
No 1
Yes No
```

Explanation

For the sample input, there is only one test case with 5 words in it and 3 strings to be checked.

`anteplop`: contains all the words from the dictionary and no extra characters. So it is both a Dictionary String and a Perfect Dictionary String. Hence, the output is `Yes Yes`.

`antelope`: the words “apple” cannot be made from this string. So it is not a Dictionary String and is missing 1 character (‘p’) to become a dictionary string. Hence the output `No 1`.

`penleantopan`: all the words of the dictionary can be made from this string but it also contains extra characters that are not required to build the words of the dictionary. So it is a Dictionary String but not a Perfect Dictionary String.