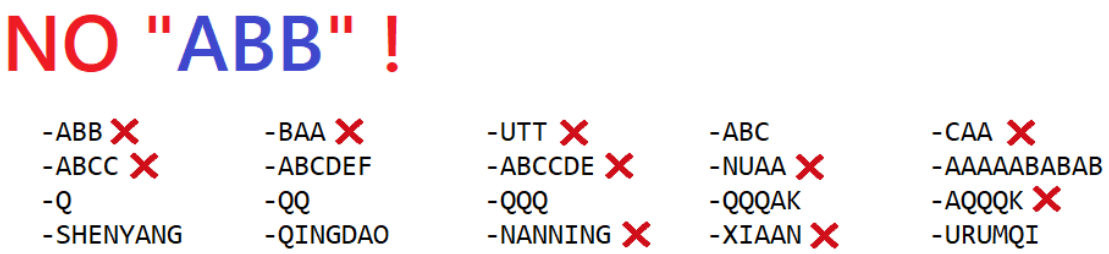
Problem D. Banned Patterns



The new censorship law has been passed and a mass of characters and strings are explicitly prohibited appearing in the public publication.

Specifically, N patterns were blacklisted. To simplify the problem, we only consider the uppercase letters and each pattern is a string that consists of only uppercase letters.

The authority makes a total ban on those strings which have at least one substring matching a banned pattern.

If a substring can be transmuted to a pattern by choosing a permutation of all 26 uppercase letters, the matching satisfies the correctness. Here we list some examples.

Each two of ABB, ACC, BAA, UTT, XZZ and ZAA match each other.

ABCDECBA matches TYQAXQYT but does not match QWEWSEWQ.

You aim to design an efficient algorithm to judge the validity of upcoming publications.

Input

The first line contains a number T ($1 \leq T \leq 20$) indicating the number of test cases.

For each test case, the first line is the number of banned patterns N ($1 \leq N \leq 5000$).

Each of the following N lines contains a pattern in capital letters.

The $(N+2)$-th line contains an integer M ($1 \leq M \leq 250000$) which is the number of strings asked to be judged based on above patterns.

Each of the following M lines consists a string in capital letters as well.

The total length of patterns in a test case should not be larger than 100000, and the total length of strings asked to be judged should not be larger than 1000000.

Output

For each test case, output a line with the index and M letters. Then for each given string, **output "Y" if it is prohibited**, or **"N" if it is allowed**.

Sample

2 2 AAB ABB 8 TUU ZZY UVW ABABABA UUVV TTUUO EEEE XY 1 ABCBD 5 UVWXY UVVVY UVWVY YVWVY VVWVY	Case #1: Y Y N N Y Y N N Case #2: N N Y N N