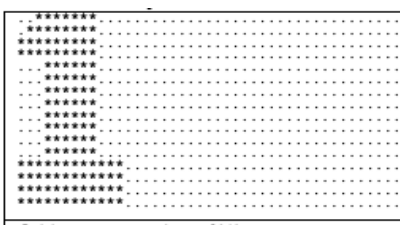
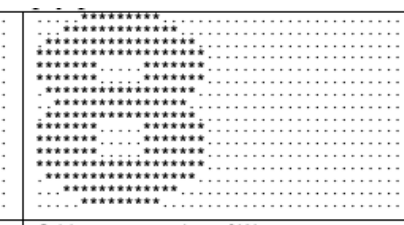
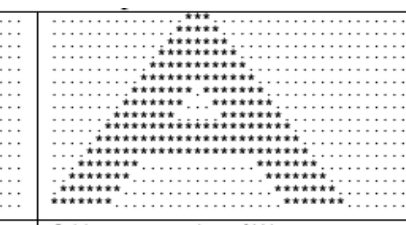
All the uppercase English letters and decimal digits (total 36 characters) can be written in (16*43) grids (16 rows and 43 columns) using dots ('.') and asterisks ('*'). Some such characters are shown below. As you can see that the empty pixels are marked with '.'s and black pixels are marked with '*'s.



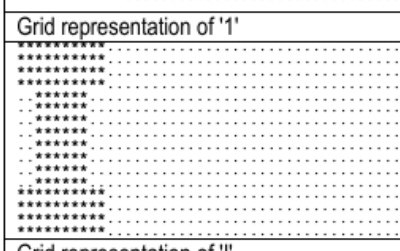
Grid representation of '1'



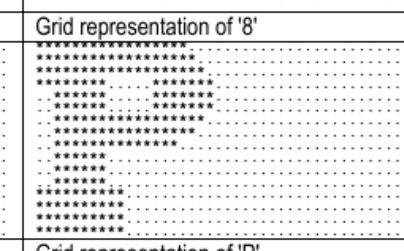
Grid representation of '8'



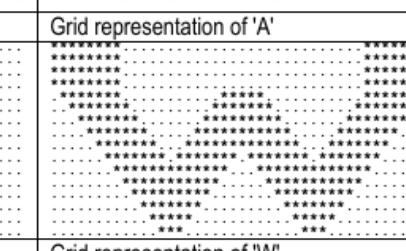
Grid representation of 'A'



Grid representation of 'I'



Grid representation of 'P'



Grid representation of 'W'

Top view of the pile when grid for character 8 is put on grid for character 1.

Top view of the pile when grid for character P is put on grid for character W.

When these characters are placed one over another (without scaling, translation or rotation) it is very difficult to judge from above how many characters are there in the pile. Sometimes it is also very difficult to say which characters are in the pile but sometimes it is very easy as you can tell whether a character is present in the pile just by checking a single pixel (if the pixel is black the character is present, otherwise absent). Suppose out of the 36 characters you have specific grid representation of specific M characters with you. From those M character grids your friend can choose any number of grids and place them in any order to form a pile and then ask you to guess which characters are present there. But you are clever and you know that there are certain characters whose presence in the pile can be detected just by checking a single pixel (regardless of whatever your friend does). Given the characters in hand your job is to identify the characters whose presence can be detected by checking only a single pixel.

Input

There is only a single set of input.

First line of the input file contains two integers N ($1 \leq N \leq 36$) and Q ($1 \leq Q \leq 1000$). Here N is the total number of character grids to consider for all the queries of this problem and Q denotes the total number of queries. The next line contains a string consisting exactly N distinct uppercase alpha-numerals. This string denotes the order in which the description alpha-numeral grids will appear in the input. The next $17N$ lines describe the grid for N characters. Each character-grid is described in exactly 16 lines but there is a line containing only empty pixels following the description of each character.

Each of the next Q lines a string of M ($1 \leq M \leq 18$) characters which describes the set of characters from which your friend can choose any number of character grids and place on a pile in any order. All the characters in this query string will be distinct.

Output

For each query produce one line of output. This line contains the serial of output followed by a string containing only 'N' or 'Y' (one for each character in the input query string). The printed character should be 'Y' if the presence of the corresponding input character grid can be identified by checking whether a single pixel is black. 'N' should be printed otherwise. (Note that sometimes if a single pixel is empty, the presence of a character can certainly be detected. But we are not considering that issue in this problem)

Sample Input

```
3 2
ABC
...............***........................
..............*****.......................
.............*******......................
............*********.....................
...........***********....................
..........*************...................
.........*******.*******..................
........*******...*******.................
.......*******.....*******................
......**********************..............
.....***********************.............
....************************............
...*******............*******...........
..*******............*******............
.*******................*******..........
*******..................*******.........
.........................................
*****************........................
******************.......................
******************.......................
********.....*******......................
..******.....*******......................
..******.....*******......................
..*****************......................
..****************.......................
..*****************......................
..******.....*******......................
..******.....*******......................
..******.....*******......................
********************.....................
********************.....................
******************......................
*****************.......................
.........................................
........*************....................
.....****************....................
...******************....................
..*******************....................
.*******.......******....................
*******..................................
*******..................................
*******..................................
*******..................................
*******..................................
*******..................................
.*******.......******....................
..*******************....................
...******************....................
.....****************....................
........*************....................
.........................................
ABC
AB
```

Sample Output

```
Query 1: YYY
Query 2: YY
```