3103 Pattern Matching

Dr. Match (Love Match) loves matching patterns to things. One day she created a simulated Artificially Intelligent robot on her laptop, to be able to recognize patterns in a collection of things. What the simulated robot does is quite simple — something called pattern matching.

What the robot was supposed to do was find entries in a list of entries (containing either words, numbers, punctuation marks, and combination of numbers and letters) that fit a given pattern. The pattern is composed of the following symbols:

`#` stands for a number from '`0`'..'`9`'

`-` stands for a punctuation mark, which is limited to the following: '`?`' '`!`' '`.`' '`,`' (question mark, exclamation point, period, comma)

`_` stands for a letter, either lower or upper case

`N` stands for null or no defined pattern

Dr. Match has already written the simulation, but she would like to know whether the robot was performing as expected — she didn't have the time to hand-check every output against the list of entries. She asked you to write a program which will validate whether or not the outputs of the robot conform to the given pattern(s).

A successfully matched pattern should not consider excess characters from the input. For example, the pattern "`###`" matches "`123456`", since the first three characters are sufficient to match.

Input

The first line of the input file consists of the number of test cases the robot performed.

For each test case,

the first line consists of the size, s, of the list of entries,

followed by s lines containing each entry,

another line for the given pattern the robot matched,

another line for the number of results, r, determined by the robot,

followed by r lines for each entry the robot matched to the given pattern.

Output

The output should consist of the case number, and whether or not the Robot performed as expected.

If the robot printed an entry which didn't match the pattern, an error indicating that the invalid result is printed:

`Robot should not print` *pattern*

If the robot did not print all the matching patterns, then the unprinted patterns should be printed:

`Robot did not print` *pattern*

If the robot was correct, the program should print:

`Robot behaved properly`

The output for '`did not print`' and '`should not print`' lines should follow the order of the input.

Sample Input

```
3
6
0io00iadfg
adfjg12daf
asdofq3442
asd-01ihfo
eoig00s.x2
sadfjg992822
______####
1
asdofq3442
3
ads123asd123
aaaa23aaa234
aaa12d
___###
2
ads123asd123
aaa12d
1
asd123
___###
1
asd123
```

Sample Output

```
Case #1:
Robot did not print sadfjg992822
Case #2:
Robot should not print aaa12d
Case #3:
Robot behaved properly
```