Xtreme9.0 - Zoom In

Some years ago, we had terminals there were capable of supporting only ASCII characters. We would like your help to construct a program, which given an input string and specific printing rules, produces the same text in a bigger layout.

Input Format

On the first line of input is an integer n, $1 \le n \le 100$, representing how many columns each character will use when printed "zoomed-in".

The next line contains an integer m, 1 <= m <= 100, representing how many rows each character will use when printed "zoomed-in". Note that n and m are not necessarily equal.

The third line contains an integer k, $3 \le k \le 95$, which indicates how many characters may need to be translated.

Following these first lines, are *k* descriptions of the "zoomed-in" characters, formatted as follows:

- On a line by itself, a single character, which has an ASCII value of between 32 and 126, inclusive
- *m* lines, each containing *n* characters, that give the "zoomed-in" representation of the character on the previous line

Following the descriptions of the zoomed in characters, is an integer number x, $1 \le x \le 500$.

Finally there are *x* lines, each containing a string of up to 2,000 characters, and ending with a new line. The characters in this string will be chosen from the set of *k* characters previously specified.

Notes:

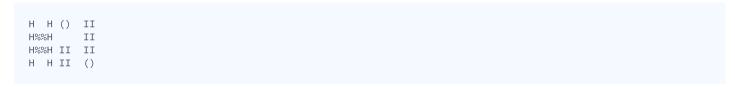
- We don't know if k sets (i.e. the descriptions of the k "zoomed-in" characters) are given in a sorted or random order.
- The "zoomed-in" version of an empty string is *m* blank lines (i.e. lines with only a newline character).

Output Format

For each of the *x* strings, you should output the "zoomed-in" version. Each string should begin on a newline.

Note: You should perform only the transformation that is specified. You should not add any space between your printed letters that is not in the transformation.

Sample Input



Explanation

For clarity, we will add dashes where the spaces would appear in the output in this explanation. According to the input, each character will use 4 rows and 4 columns, and there are 3 characters that may be translated.

A capital H ('H') should be translated as

```
H--H
H%%H
H%%H
H--H
```

A lower-case i ('i') should be translated as

```
-II-
-II-
-()-
```

An exclamation mark ('!') should be translated as:

```
-II-
-II-
-II-
-()-
```

We are then asked to print the "zoomed in" version of the string "Hi!". The output would be the following (with spaces where the dashes are located):

```
H--H-()--II-
H%%H----II-
H%%H-II--II-
H--H-II--()-
```