Code Jam 2009 - Round 1B

Square Math

Problem

Say we have a square that has **W** cells on each side and, therefore, $\mathbf{W}^2$ cells total. Let's go further and fill each cell with one of the following:

- A digit from 0 to 9;
- The addition sign (+);
- The subtraction sign (-).

If, finally, we add a constraint that no 2 digits are horizontally or vertically adjacent and no 2 operators (+ or -) are horizontally or vertically adjacent, then our square can be called an "arithmetic square".

Square Math is the name of a puzzle where, given an arithmetic square, we start from any numeric cell and move either horizontally or vertically a cell at a time, finally ending in a numerical cell. The mathematical expression we get from the traversal is evaluated to get a single value. For example:

```
2+3
+4-
1+0
```

The above is a valid arithmetic square of size **W** = 3. If we start from "2", move horizontally right, then vertically down, we'll get "2+4", which gives a value of "6". If we further move horizontally right, then vertically up, we'll get "2+4-3", which is equal to "3".

In Square Math, there is no limit to how many times you can use a particular cell. It is perfectly legal to move from a cell to its neighbor, then back to the original cell. Given an arithmetic square and a list of queries, your task is to find a Square Math expression which evaluates to each query.

Input

The first line of input contains a single integer, **T**. **T** test cases follow. The first line of each test case contains 2 integers, **W** and **Q**. **W** lines follow, each containing **W** characters, representing the arithmetic square. Don't worry, all arithmetic squares in the input are well-formed. The following line contains a space separated list of **Q** integers, representing the values which need to be computed by using Square Math (the queries). You can assume that all given values will have at least one possible Square Math solution.

Output

For each test case, begin output with "Case **#X**:" on a line by itself, where **X** is the test case number, starting from 1. Then, for each query within the test case, print the Square Math expression which evaluates to the query on a line by itself.

In the case where there are multiple possible Square Math expressions, print the one that is shortest. If there is still a tie, print the lexicographically smallest expression. Remember that '+' is lexicographically smaller than '-'.

Limits

Memory limit: 1 GB.
$1 \le \mathbf{T} \le 60$

Small dataset

Time limit: 30 seconds.
$2 \le \mathbf{W} \le 10$
$1 \le \mathbf{Q} \le 20$
1 ≤ each query ≤ 50

Large dataset

Time limit: 60 seconds.
$2 \le \mathbf{W} \le 20$
$1 \le \mathbf{Q} \le 50$
1 ≤ each query ≤ 250

Sample

Sample Input

```
2
5 3
2+1-2
+3-4+
5+2+1
-4-0-
9+5+1
20 30 40
3 2
2+1
+4+
5+1
2 20
```

Sample Output

```
Case #1:
1+5+5+9
3+4+5+9+9
4+9+9+9+9
Case #2:
2
5+5+5+5
```