And the turtles, of course...all the turtles are free
As turtles, and, maybe, all creatures should be.

Of course, the turtles aren't completely free. In particular, they aren't free to swim where other turtles are swimming. To learn more about the swimming behaviour of turtles you decide to write a computer simulation of a pond and the turtles swimming in it.

For simplicity, you represent the pond as a rectangular grid, with turtles occupying positions at points on the grid. If the pond is an $N \times M$ grid, a position on the grid may be represented by a pair of integer coordinates (i, j), with $0 \leq i < N$ and $0 \leq j < M$. The grid is aligned with its first dimension running north-south and its second dimension running east-west. Coordinate values increase to the south and the east.

A turtle swimming in the pond may be requested to move to the adjacent grid position in one of eight directions: N, S, E, W, NE, NW, SE, SW. If the request would cause the turtle to move off the grid or cause it to move onto a grid position occupied by another turtle, the request is ignored. Otherwise, the turtle happily obeys the movement request. (Turtles are easy to push around.)

Input

The input consist of several data sets. Each data set begins with a description of the pond and the initial location of the turtles. The first line of the input consists of four integers separated by one or more spaces. The first integer specifies the size of the pond in the north-south direction (N), the second integer specifies the size of the pond in the east-west direction (M), the third integer specifies the number of turtles in the pond (T) and the fourth integer indicates the number of turtle movement requests (K). This first line is followed by T lines, each providing information on a single turtle. Each line of turtle information consists of three integers separated by one or more spaces. The first integer specifies a turtle id, the second integer specifies an initial grid position for the turtle along the north-south dimension, and the third integer specifies an initial grid position for the turtle along the east-west dimension. All turtles will be located at valid grid positions; no two turtles will be located at the same grid position; turtle ids are in the range one to ten thousand; the maximum size of the grid in either dimension is sixty; the minimum size is two.

Following the description of the pond and its initial turtle configuration, the remainder of the input, consists of a sequence of K turtle movement requests, one per line. Each movement request consists of a turtle id followed by a movement direction, and indicates that the specified turtle should be requested to move one grid position in the specified direction. The turtle id and movement direction are separated by one or more spaces. The movement direction is one of: '`N`', '`S`', '`E`', '`W`', '`NE`', '`NW`', '`SE`', '`SW`'. The turtle requests should be processed sequentially starting from the initial pond configuration given in the input.

Output

The output is a graphical representation of the pond configuration for each data set after all turtle movement requests have been processed. A picture of the pond is rendered using ASCII characters. Beginning with the most northerly row, each row of the grid is represented by a single line in the output. Grid positions along a row are represented by character positions in the output line. The final position of a turtle is marked with an asterisk ('`*`'); space characters are used to fill empty grid positions. If the output is displayed as text on a computer screen or printed page, the result is a map of turtle positions in the pond, with north toward the top, east toward the right, and west toward the left. Spaces should be generated only when required for positioning turtles; output lines should not have trailing spaces.

Print a blank line after each data set.

Sample Input

```
4 4 3 4
1 0 0
2 0 2
3 3 3
1 S
2 W
2 W
1 SE
4 4 3 4
1 0 0
2 0 2
3 3 3
2 W
2 W
1 S
1 SW
```

Sample Output

```
*

 *
   *

 *
*

   *
```