Problem 6—Word Crossing

After getting sick of movies, you decide to get out your well-worn puzzle book and do the word crossing games. These are a lot like crossword puzzles; however, instead of clues, you get the words themselves (in no particular order), and you have to put them into the grid. You are to write a program to do exactly that.

INPUT SPECIFICATION. Each input case will begin with two unsigned positive decimal integers, representing the height (h) and width (w) of the grid, separated by exactly one space and followed by **<EOLN>**. This will be followed by h rows of data, each containing w characters. Each row is followed by **<EOLN>**. Each character in the data will be either an asterisk (representing a letter) or a space (representing a spot where a letter cannot go). Following these rows will be an unsigned positive decimal integer, followed by **<EOLN>**, representing the number of words (n) that will go into the grid. The next n rows of data will be the words themselves, each of length at least three and containing only capital letters and each followed by **<EOLN>**. The last input case will be followed by "0 0**<EOLN>**".

OUTPUT SPECIFICATION. The output cases should appear in the same order as the input cases. Each output case will be "Case c" (where c is the case number) followed by 2 **<EOLN>** characters. Then follows h rows of w characters each representing the grid with the words filled in. An extra **<EOLN>** follows each output case. Each input case is guaranteed to yield a unique solution.

SAMPLE INPUT.
```
5•3<EOLN>                          *********•*********<EOLN>
••*<EOLN>                          33<EOLN>
••*<EOLN>                          BOHR<EOLN>
***<EOLN>                          BORN<EOLN>
••*<EOLN>                          LAMB<EOLN>
••*<EOLN>                          NEEL<EOLN>
2<EOLN>                            RYLE<EOLN>
MOP<EOLN>                          TAMM<EOLN>
HAPPY<EOLN>                        BETHE<EOLN>
18•18<EOLN>                        BLOCH<EOLN>
*******•••*******<EOLN>            DIRAC<EOLN>
*•••••••*••••*••••*<EOLN>          ESAKI<EOLN>
*•••••*********•****<EOLN>         RUSKA<EOLN>
******•*•*••*••*•*<EOLN>           SEGRE<EOLN>
*••*••*••••*••*••*••<EOLN>         STARK<EOLN>
*••*••*••••*•*******<EOLN>         STERN<EOLN>
•••*••*******•*••*•*<EOLN>         ALFVEN<EOLN>
*****•••*•*••*••*•*<EOLN>          FOWLER<EOLN>
*•••••*•*•••*••••*•*<EOLN>         LANDAU<EOLN>
*•••••*•*•*******•*•*<EOLN>        TOWNES<EOLN>
*•*****•*•••*••••••*<EOLN>         DEHMELT<EOLN>
*•*•••*•*•••*••*•••*<EOLN>         FEYNMAN<EOLN>
*****••*•••*******<EOLN>           GIAEVER<EOLN>
*•*•••••*••••••*•••*<EOLN>         KAPISTA<EOLN>
*•*••*********•*****<EOLN>         KASTLER<EOLN>
*•*••••••*••*••*•••*<EOLN>         KENDALL<EOLN>
••*•••••••*••*••*•••<EOLN>         ANDERSON<EOLN>
```

```
BLACKETT<EOLN>                              BECQUEREL<EOLN>
EINSTEIN<EOLN>                              MOTTELSON<EOLN>
FRIEDMAN<EOLN>                              0•0<EOLN>
LEDERMAN<EOLN>                              <EOF>
ROENTGEN<EOLN>
TOMONAGA<EOLN>
```

SAMPLE OUTPUT.

```
Case•1<EOLN>
<EOLN>
••H<EOLN>
••A<EOLN>
MOP<EOLN>
••P<EOLN>
••Y<EOLN>
<EOLN>
Case•2<EOLN>
<EOLN>
FRIEDMAN•••KENDALL<EOLN>
O••••••E••••I••••A<EOLN>
W••••ROENTGEN•TAMM<EOLN>
LANDAU•L•O••S••N•B<EOLN>
E••I•S•••W••T••D••<EOLN>
R••R•K•••N•DEHMELT<EOLN>
•••A•ALFVEN•I••R•O<EOLN>
BLOCH••E•S••N••S•M<EOLN>
E••••B•Y•••S•••O•O<EOLN>
C••••O•N•BETHE•N•N<EOLN>
Q•BOHR•M•••A•••••A<EOLN>
U•L••N•A•••R••G••G<EOLN>
ESAKI••N•••KAPISTA<EOLN>
R•C•••••S•••••A•••<EOLN>
E•K••KASTLER•SEGRE<EOLN>
L•E•••••E••Y••V•••<EOLN>
••T•••••R••L••E•••<EOLN>
MOTTELSON•LEDERMAN<EOLN>
<EOLN>
<EOF>
```