Problem 6—Word Brick

Professor Farnsworth enjoys playing word games, especially the word search from the daily newspaper. However, those two dimensional grids are too banal for him to bother with. Farnsworth prefers to search for words in three-dimensional bricks of letters.

Given a brick of letters, you need to locate a given word in it. The word may appear orthogonally or diagonally, forward, backward, up, down, front, back, whatever. The letters of the words are in cubes that touch each other and are arranged linearly (the word doesn't change direction midword).

INPUT SPECIFICATION. Each input case consists of three integers, separated by one space and followed by two **<EOLN>**'s. These numbers are, respectively, the number of levels of the brick, the number of rows in each level, and the number of columns in each level. Then follows the brick, one level at a time, with an **<EOLN>** following each row and an extra **<EOLN>** following each level. The brick consists entirely of capital letters. "0 0 0**<EOLN>"** follows the last case and is not processed.

OUTPUT SPECIFICATION. The output cases appear in the same order as their corresponding input cases. Each output case is of the form "Case *c*: *w* was found at (*x*,*y*,*z*)" or "Case *c*: *w* was not found.", followed by **<EOLN>,** where *c* is the case number, and *w* is the word, and *x*, *y*, and *z* are respectively the level number, the row number, and the column number of the first letter of the word searched for. All coordinates are numbered starting from 1 with (1,1,1) corresponding to the first letter in the brick to appear in the input case. If the search word appears in the brick at all, it only appears once. An extra **<EOLN>** follows each output case.

SAMPLE INPUT.

```
1·1·3<EOLN>
<EOLN>
DOG<EOLN>
<EOLN>
GOD<EOLN>
<EOLN>
3·1·1<EOLN>
<EOLN>
D<EOLN>
<EOLN>
O<EOLN>
<EOLN>
G<EOLN>
<EOLN>
CAT<EOLN>
<EOLN>
3·3·3<EOLN>
<EOLN>
QWT<EOLN>
FGH<EOLN>
JKL<EOLN>
<EOLN>
ERT<EOLN>
UAW<EOLN>
XYZ<EOLN>
<EOLN>
BBB<EOLN>
BBB<EOLN>
BBB<EOLN>
<EOLN>
BAT<EOLN>
<EOLN>
0·0·0<EOLN>
<EOF>
```

SAMPLE OUTPUT.

```
Case·1:·GOD·was·found·at·(1,1,3)<EOLN>
<EOLN>
Case·2:·CAT·was·not·found.<EOLN>
<EOLN>
Case·3:·BAT·was·found·at·(3,3,1)<EOLN>
<EOLN>
<EOF>
```