Scrabble

Computer Science Society
Programming Contest
Winter 2011

Scrabble is a game where players try to form words using letters from the alphabet $\{\mathtt{a}, \mathtt{b}, \mathtt{c}, \ldots, \mathtt{z}\}$. The letters are printed on square wooden tiles that can occupy one cell of an $n \times n$ grid of cells known as the *board*. A cell position is an ordered pair (x, y), where $0 \leqslant x < n$ is a column index and $0 \leqslant y < n$ is a row index; the lower left cell position is $(0, 0)$. We will assume the board is clear before placing each word and the first letter can occupy any cell (x, y). Successive letters must occupy adjacent cells, either horizontally *across* (from left to right) or vertically *down* (from top to bottom), and must fit within the board. A word $w = w_1 w_2 \ldots w_k$ whose first letter is placed at position (x, y) earns a numeric score, depending on the integer value $v(w_i)$ of each letter w_i (e.g. $v(\mathtt{a}) = 1$, $v(\mathtt{h}) = 4$, $v(\mathtt{z}) = 10$), and on letter multipliers $L(x, y) \in \{1, 2, 3\}$ and word multipliers $W(x, y) \in \{1, 2, 3\}$ assigned to each cell position. Specifically, the score of a word $w = w_1 w_2 \ldots w_k$ placed at position (x, y) is

$$\text{score} = \Big(\prod_{w \text{ covers } (x,y)} W(x, y) \Big) \cdot \Big(\sum_{w_i \text{ is at } (x,y)} v(w_i) \cdot L(x, y) \Big).$$

Input Format

The first line of input contains the letter values $v(\mathtt{a}), v(\mathtt{b}), v(\mathtt{c}), \ldots, v(\mathtt{z})$ separated by blanks. The remaining lines of input describe one or more board scenarios, each followed by an empty line. The first line of a board scenario describes the board size $n > 0$. The next four lines of a board scenario contain the lists of positions such that $L(x, y) = 2$, $L(x, y) = 3$, $W(x, y) = 2$, and $W(x, y) = 3$, respectively. The remaining lines of a board scenario each contain a word $w = w_1, w_2 \ldots w_k$, a position (x, y) where the first letter w_1 is placed, and the direction of the word (`down` or `across`).

Output Format

For each board scenario, output the score of each word as shown in the sample output. Assume the board is clear before placing and scoring each word.

Input Sample

```
1 3 3 2 1 4 2 4 1 8 5 1 3 1 1 3 10 1 1 1 1 4 4 8 4 10
3
2 0  0 2
1 0  2 1
1 1  2 2
0 0
bug 0 0 across
zap 0 2 down

2
0 0

1 1

if 0 1 down
ox 0 1 across
```

Output Sample

```
3x3 board:
bug at (0,0) across scores 30
zap at (0,2) down scores 72

2x2 board:
if at (0,1) down scores 9
ox at (0,1) across scores 18
```