F. Fruit burst

Program:	`fruit.(cpp|java|py)`
Input:	`fruit.in`
Balloon Color:	`Black`

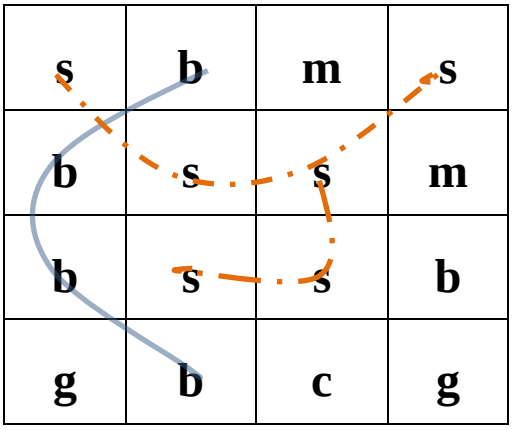
Description

Kids love to play fruit burst. It is a board game where the board is an *L* x *L* square grid, and each grid cell contains a fruit. A player connects at least *n* fruits of the same type, and bang! All those connected fruits burst, and he gets *p* * *k* points, where *p* is the point for the fruit and *k* is the number of fruits (same type) that the player connected. Your task is to find the highest point possible given the current board state.

Example

As an example, consider the following board (4x4 grid) and 5 types of fruits, *s*, *b*, *g*, c, and *m* having 10, 16, 25, 10, and 10 points, respectively. Also, *n* = 3 in this board. Two fruits are connected if they share a common edge or a corner in the grid. The board shows that maximum six *s* fruits can be connected and maximum four *b* fruits can be connected. So, when the player connects *s*, all *s* fruits burst and he gets 60 points. If he connected four *b* fruits, he would get 64. Note that *c*, *g* and *m* fruits do not burst because the player can only connect less than 3 such fruits. Therefore, they don't give any point. Therefore, the max burst is 64.

Fruit points: b = 16, g = 25 and default = 10 (i.e., s = c = m = 10)



burst(b) = 4 x 16 = 64 (connected b's shown in solid line)

burst(s) = 6 x 10 = 60 (connected s's shown in the dotted line)

burst(c) = burst(g) = burst(m) = 0

Input

The input consists of several test cases. The first line of each test case contains L (2 < L <= 100) and *n* (1 < n < *L*) followed by the grid description. The grid is given in *L* lines, each line containing *L* fruit names separated by space(s). A fruit name is a lower case character [a-z]. After the grid input, fruit

points are given in one line. In this line, a fruit name is given followed by its points, separated by spaces. The last entry in this line will be the character '#' followed by the default fruit points. Any fruit not mentioned in this line will assume the default points. All fruit points are integers > 0. Input is terminated by a case having $K = 0$, which should not be processed.

Output

For each test case, you are to output one line, containing the max burst value.

Sample Input / Output

fruit.in

```
4 3
s b m s
b s s m
b s s b
g b c g
g 25 b 16 # 10
0
```

OUTPUT

```
64
```