2019 Canadian Computing Olympiad
Day 1, Problem 1
Human Error

Time Limit: 1 second

Problem Description

Justin and Donald are playing their favourite game: hop chess. You probably haven't heard of it, but the rules are pretty simple.

The board is a rectangular grid, with each square of the board initially having exactly one player's piece on it. Justin's pieces are denoted as `J`, with Donald's being `D`. Each player has at least one piece on the grid initially.

The game begins with Justin playing first. On a turn, a player may move one of his pieces to capture (and thus remove from the board) a neighbouring piece (not necessarily the opponent's). A piece X is said to be neighbouring Y if it is either up, down, left, or right of Y. If such a move cannot be made, then the active player loses.

In an ideal world, both Justin and Donald are perfect logicians, and capable of discerning an optimal strategy for any board. Then perhaps we might be interested in who of the two would win. But that wouldn't be very realistic. Indeed, when playing, Justin and Donald can both come up with a relatively good solution; exactly how good it tends to be is determined by their error factors, J and D respectively.

Formally, the active player with error factor A first chooses a *proposal set*: either the set of all possible moves if there are A or less possible moves or some subset of size A from the set of possible moves if he has more than A possible moves. Then, from this *proposal set*, the player selects a move randomly with equal probability.

Both players, when given the choice of choosing their *proposal set*, chooses the most optimal such set (one which produces the highest probability of winning), knowing that the other player always chooses their *proposal set* optimally as well.

The natural question is then: exactly what is the probability that Justin wins a game of hop chess, given the initial board, J, and D?

Input Specification

Input will begin with two space-separated positive integers R, C $(R \cdot C \leq 13)$. On the next R lines will be strings of C characters drawn from the set $\{$`J`, `D`$\}$, describing the initial board state. Finally, there will be two space-separated integers, J, D $(1 \leq J, D \leq 13)$

Output Specification

Output a single floating point number rounded to 3 decimal places: the probability that Justin wins.

Sample Input 1

```
1 3
JJD
3 1
```

Output for Sample Input 1

```
0.667
```

Explanation of Output for Sample Input 1

Note that Justin has 3 possible moves (note that _ indicates an empty cell in all explanations below):

- he moves his first piece right, capturing his second piece, and ensuring his loss by having the board appear as _JD;
- he moves his second piece right, capturing Donald's piece and securing victory, with the board as J_J;
- or he moves his second piece left, capturing his own piece, but leaving Donald unable to move, thus also winning, with the board as J_D.

Clearly the latter 2 cases are optimal—but since Justin has error factor 3, there is a 1/3 chance that he chooses the option causing him to lose. Thus he wins with probability 2/3.

Sample Input 2

```
2 2
JJ
DD
3 1
```

Output for Sample Input 2

```
0.000
```

Explanation of Output for Sample Input 2

There is no hope for Justin to win.

To see why, notice that Justin has 4 possible first moves:

```
J_    _J    J_    _J
DD    DD    DJ    JD
```

He can pick any subset of size three from the above moves.

However, Donald will always pick his most optimal move. Regardless of Justin's first move, Donald will leave the board in one of the following configurations:

```
D_      _J      _D      J_
_D      D_      D_      _D
```

all of which will cause Justin to lose.