Problem A. **Monster Path**

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the Quick-Start Guide to get started.

Small input 7 points	Solve A-small
Large input 8 points	Solve A-large

Problem

Codejamon is a mobile game in which monster trainers walk around in the real world to catch monsters. You have an old smartphone with a short battery life, so you need to choose your path carefully to catch as many monsters as possible.

Suppose the Codejamon world is a rectangular grid with **R** rows and **C** columns. Rows are numbered from top to bottom, starting from 0; columns are numbered from left to right, starting from 0. You start in the cell in the $\mathbf{R_s}$th row and the $\mathbf{C_s}$th column. You will take a total of **S** unit steps; each step must be to a cell sharing an edge (not just a corner) with your current cell.

Whenever you take a step into a cell in which you have not already caught a monster, you will catch the monster in that cell with probability **P** if the cell has a monster attractor, or **Q** otherwise. If you do catch the monster in a cell, it goes away, and you cannot catch any more monsters in that cell, even on future visits. If you do not catch the monster in a cell, you may still try to catch the monster on a future visit to that cell. The starting cell is special: you have no chance of catching a monster there before taking your first step.

If you plan your path optimally before making any move, what is the maximum possible expected number of monsters that you will be able to catch?

The battery can only support limited steps, so hurry up!

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow.

Each test case starts with a line of five integers: **R**, **C**, $\mathbf{R_s}$, $\mathbf{C_s}$ and **S**. **R** and **C** are the numbers of rows and columns in the grid; $\mathbf{R_s}$ and $\mathbf{C_s}$ are the numbers of the row and column of your starting position, and **S** is the number of steps you are allowed to take.

The next line contains two decimals **P** and **Q**, where **P** is the probability of meeting a monster in cells with a monster attractor, and **Q** is the probability of meeting a monster in cells without a monster attractor. **P** and **Q** are each given to exactly four decimal places.

Each of the next **R** lines contains contains **C** space-separated characters; the j-th character of the i-th line represents the cell at row i and column j. Each element is either `.` (meaning there is no attractor in that cell) or `A` (meaning there is an attractor in that cell).

Output

For each test case, output one line containing `Case #x: y`, where x is the test case number (starting from 1) and y is the largest possible expected number of monsters that the player can catch in the given amount of steps.

y will be considered correct if y is within an absolute or relative error of 10^{-6} of the correct answer. See the FAQ for an explanation of what that means, and what formats of real numbers we accept.

Limits

$1 \le \mathbf{T} \le 100$.
$0 \le \mathbf{R_s} < \mathbf{R}$.
$0 \le \mathbf{C_s} < \mathbf{C}$.
$0 \le \mathbf{Q} < \mathbf{P} \le 1$.

Small dataset

$1 \le \mathbf{R} \le 10$.
$1 \le \mathbf{C} \le 10$.
$0 \le \mathbf{S} \le 5$.

Large dataset

$1 \le \mathbf{R} \le 20$.
$1 \le \mathbf{C} \le 20$.

$0 \le \mathbf{S} \le 9$.

Sample

Input

```
2
4 4 0 0 5
0.8000 0.2000
. . . .
. . . .
. . A .
. A . A
10 10 9 1 4
0.6121 0.1000
. . A A . . . . . .
A . . . . . . . . .
. . A . . . . A . .
. . . A A . . . . .
. A A A . . . . . A
A . A A . . . . A .
. A . . . . . A . .
. . . . A A . . . .
. . A . . . A . . A
. . . . A . . A . .
```

Output

```
Case #1: 1.6000000
Case #2: 1.0495336
```

In Case #1, one of the best paths is (0,0)->(0,1)->(0,2)->(1,2)->(2,2)->(2,3). On this path, the expected number of monsters that you will catch is 0.2 + 0.2 + 0.2 + 0.8 + 0.2 = 1.6. Remember that there is no chance of catching a monster before taking your first step, which is why there are five probabilities in the calculation, not six.

In Case #2, one of the best paths is (9,1)->(9,2)->(8,2)->(8,3)->(8,2). On this path, the expected number of monsters that you will catch is 0.1 + 0.6121 + 0.1 + 0.23743359 = 1.04953359. Since we accept results within an absolute or relative error of 10^{-6} of the correct answer (1.04953359), 1.0495336 is accepted.