

# Funniest Word Search

## Problem

Siv's birthday is next week and Cel is preparing a birthday present for her. Because Siv loves puzzles, Cel is creating a word search puzzle as a gift.

In a word search puzzle, the solver is given a rectangular grid with **R** rows and **C** columns, and the solver must find all valid words hidden inside the grid. Each hidden word may appear horizontally or vertically (but NOT diagonally) in the grid, forward or in reverse. Hidden words may overlap.

Cel has a dictionary with **W** different words; these are the only words that can be hidden within the puzzle grid. (Of course, not every contiguous horizontal or vertical part of the grid will necessarily contain one of the hidden words.) These words are not necessarily real English words. Each word might appear in the grid one or more times, or not at all.

Cel has already created the word search puzzle. However, there is a problem: the puzzle is too big to print on a sheet of paper. As Siv's birthday is coming soon, there is not enough time to create a new word search puzzle from scratch. So Cel is wondering whether she can reduce the size of the grid, simply by selecting a non-empty, grid-aligned subgrid from the original grid.

Randomly selecting a subgrid might result in a boring puzzle without a lot of hidden words. So Cel wants to select a subgrid with the largest *fun value*, where the *fun value* of a subgrid is defined as:

$$\text{fun value} = (\text{total length of words matched}) / ((\text{width of subgrid}) + (\text{height of subgrid}))$$

Notes:

- The entire word has to appear in the subgrid to be counted.
- If a word appears **x** times in the subgrid, its length should be added **x** times in the above formula.
- If a word and its reverse both appear in the subgrid (even at the same position!), we count both occurrences.
- The subgrid with largest *fun value* might be the entire original grid.

Can you please help Cel find the largest *fun value* that a subgrid could have, and the number of different subgrids that have this *fun value*? Two subgrids are considered different if and only if there is some cell in the grid—that is, some (row, column) position—that is in one subgrid, but not the other.

## Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. Each test case is structured as follows:

- The first line contains 3 integers **R**, **C** and **W**, as described above.
- Each of the next **R** lines contains exactly **C** uppercase English letters.
- Each of the next **W** lines contain exactly one valid word. Each word contains only uppercase English letters.

## Output

For each test case, output one line containing `Case #x: y/z n`, where:

- $x$  is the test case number (starting from 1).
- $y/z$  is an irreducible fraction equal to the largest possible *fun value* (as described above) of a subgrid.
- $y$  is a non-negative integer.
- $z$  is a positive integer.
- $n$  is the number of subgrids with *fun value* equal to  $y/z$ .

$y/z$  is called an irreducible fraction if the greatest common divisor of  $y$  and  $z$  is 1.

## Limits

- $1 \leq T \leq 100$ .
- Time limit: 240 seconds per test set.
- Memory limit: 1 GB.
- $1 \leq R \leq 100$ .
- $1 \leq C \leq 100$ .
- No word appears more than once in the list of valid words.
- The combined length of all words in the list of valid words is at most 5000 letters.

## Small dataset (Test set 1 - Visible)

- The length of each valid word is exactly 1.
- $1 \leq W \leq 26$ .

## Large dataset (Test set 2 - Hidden)

- $1 \leq W \leq 1000$ .

## Sample

*Note: there are additional samples that are not run on submissions down below.*

### Sample Input

```
2
1 2 1
AA
A
1 2 1
AA
B
```

### Sample Output

```
Case #1: 8/3 1
Case #2: 0/1 3
```

Let  $(i, j)$  denotes the cell at  $i$ -th row and  $j$ -th column.

In Sample Case #1, the subgrid with highest *fun value* is the entire grid. The valid word `A` appears 8 times in the grid:

- 2 times horizontally (once forward and once in reverse) at cell  $(1, 1)$ .
- 2 times vertically (once forward and once in reverse) at cell  $(1, 1)$ .
- 4 times (horizontally and vertically, forward and reverse) at cell  $(1, 2)$ .

In Sample Case #2, there are 3 subgrids with *fun value* equals 0:

- Subgrid consists of only one cell (1, 1).
- Subgrid consists of only one cell (1, 2).
- Subgrid with top-left corner at cell (1, 1) and bottom-right corner at cell (1, 2).

## Additional Sample - Test Set 2

*The following additional sample fits the limits of Test Set 2. It will not be run against your submitted solutions.*

Sample Input	Sample Output
<pre> 2 1 3 2 ABC ABC CBA 4 4 1 AAAB AAAB AAAB BBBB AA </pre>	<pre> Case #1: 3/2 1 Case #2: 8/1 1 </pre>

In Sample Case #1, the subgrid with highest *fun value* is the entire grid. The valid word `ABC` appears horizontally from cell (1, 1), and the valid word `CBA` appears horizontally in reverse from cell (1, 1). So the *fun value* of the entire grid equals  $6 / (1 + 3) = 3/2$ .

In Sample Case #2, the subgrid with highest *fun value* has top-left corner at cell (1, 1) and bottom-right corner at cell (3, 3). The valid word `AA` appears 24 times in the subgrid:

- 12 times horizontally (6 times forward and 6 times in reverse).
- 12 times vertically (6 times forward and 6 times in reverse).

**Note:** We do not recommend using interpreted/slower languages for this problem.