Submissions

How Big Are the Pockets?

5pt Not attempted **311/426 users** correct (73%)

15pt Not attempted **195/249 users** correct (78%)

Portal

10pt Not attempted **159/207 users** correct (77%)

15pt Not attempted **123/163 users** correct (75%)

No Cheating

10pt Not attempted **510/637 users** correct (80%)

20pt Not attempted **68/114 users** correct (60%)

Endless Knight

5pt Not attempted **845/879 users** correct (96%)

20pt Not attempted **32/173 users** correct (18%)

Top Scores

bmerry	100
yuhch123	100
halyavin	85
wata	80
iwi	80
Ahyangyi	80
tourist	80
gawry	80
vlad89	80
neal.wu	80

Problem A. How Big Are the Pockets?

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.

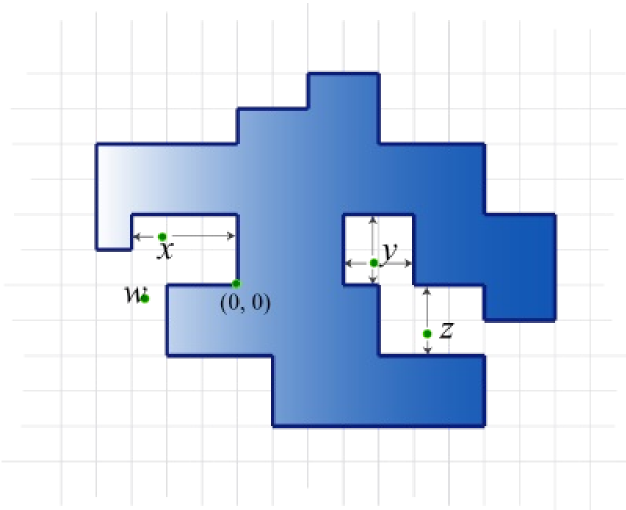
Small input 5 points	Solve A-small
Large input 15 points	Solve A-large

Problem

Professor Polygonovich, an honest citizen of Flatland, likes to take random walks along integer points in the plane. He starts from the origin in the morning, facing north. There are three types of actions he makes:

- 'F': move forward one unit of length.
- 'L': turn left 90 degrees.
- 'R': turn right 90 degrees.

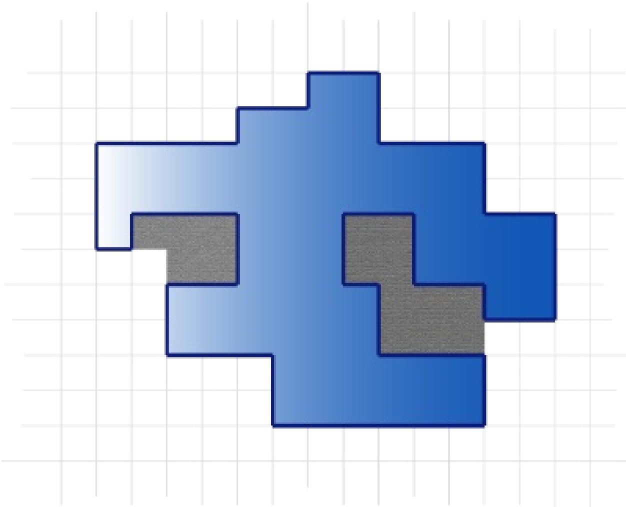
At the end of the day (yes, it is a long walk!), he returns to the origin. He never visits the same point twice except for the origin, so his path encloses a polygon. In the following picture the interior of the polygon is colored blue (ignore the points x, y, z, and w for now; they will be explained soon):



Notice that as long as Professor Polygonovich makes more than 4 turns, the polygon is not convex. So there are pockets in it.

Warning! To make your task more difficult, our definition of *pockets* might be different from what you may have heard before.

The gray area below indicates pockets of the polygon.



Formally, a point **p** is said to be in a pocket if it is not inside the polygon, and at least one of the following two conditions holds.

- There are boundary points directly both east and west of **p**; or
- There are boundary points directly both north and south of **p**.

Boundary points are the points traversed by Mr. Poligonovich on his walk (these include all points, not just those with integer coordinates).

Consider again the first picture from above. Point **x** satisfies the first condition; **y** satisfies both; **z** satisfies the second one. All three points are in pockets. The point **w** is not in a pocket.

Given Polygonovich's walk, your job is to find the total area of the pockets.

Input

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

Each test case has the description of one walk of Professor Polygonovich. It starts with an integer **L**. Following are **L** "**S T**" pairs, where **S** is a string consisting of 'L', 'R', and 'F' characters, and **T** is an integer indicating how many times **S** is repeated.

In other words, the input for one test case looks like this:

$$S_1\ T_1\ S_2\ T_2\ ...\ S_L\ T_L$$

The actions taken are the concatenation of T_1 copies of S_1, followed by T_2 copies of S_2, and so on.

The "**S T**" pairs for a single test case may not all be on the same line, but the strings **S** will not be split across multiple lines. The second example below demonstrates this.

Output

For each test case, output one line containing "Case #**X**: **Y**", where **X** is the 1-based case number, and **Y** is the total area of all pockets.

Limits

1 ≤ **N** ≤ 100
1 ≤ **T** (bounded from above by constraints in the problem statement, "Small dataset" and "Large dataset" sections)
The path, when concatenated from the input strings, will not have two consecutive direction changes (that is, there will be no 'LL', 'RR', 'LR', nor 'RL' in the concatenated path). There will be at least one 'F' in the path.
The path described will not intersect itself, except at the end, and it will end back at the origin.

Small dataset

1 ≤ **L** ≤ 100
The length of each string **S** will be between 1 and 16, inclusive.
The professor will not visit any point with a coordinate bigger than 100 in absolute value.

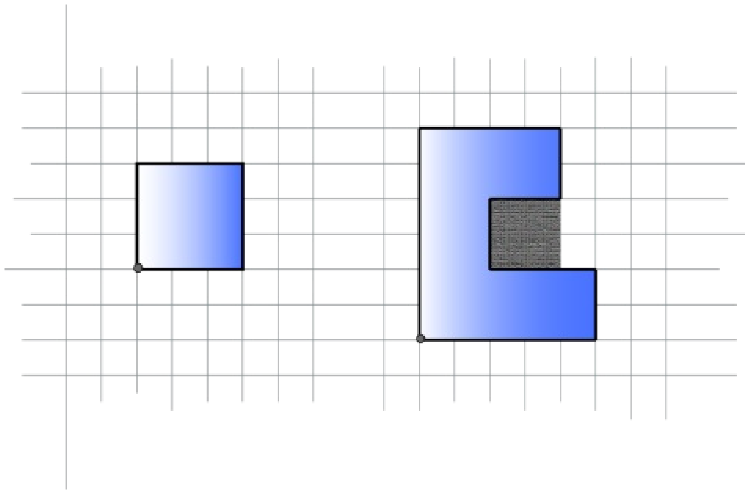
Large dataset

1 ≤ **L** ≤ 1000
The length of each string **S** will be between 1 and 32, inclusive.
The professor will not visit any point with a coordinate bigger than 3000 in absolute value.

Sample

Input	Output
2 1 FFFR 4 9 F 6 R 1 F 4 RFF 2 LFF 1 LFFFR 1 F 2 R 1 F 5	Case #1: 0 Case #2: 4

The following picture illustrates the two sample test cases.



Problem B. **Portal**

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 10 points	Solve B-small
Large input 15 points	Solve B-large

Problem

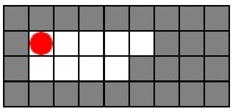
Portal[TM] is a first-person puzzle/platform game developed and published by Valve Software. The idea of the game was to create two portals on walls and then jump through one portal and come out the other. This problem has a similar idea but it does not assume you have played Portal.

For this problem you find yourself in a **R** by **C** grid. Additionally there is a delicious cake somewhere else in the grid. You're very hungry and wish to arrive at the cake with as few moves as possible. You can move north, south, east or west to an empty cell. Additionally, you have the ability to create portals on walls.

To help you get to the cake you have a portal gun which can shoot two types of portals, a yellow portal and a blue portal. A portal is created by shooting your portal gun either north, south, east or west. This emits a ball of energy that creates a portal on the first wall it hits. Note that for this problem shooting the portal gun does not count as a move. If you fire your portal gun at the cake, the energy ball will go right through it.

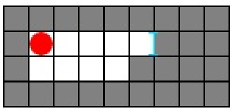
After creating a yellow portal and a blue portal, you can move through the yellow portal to arrive at the blue portal or vice versa. Using these portals you may be able to reach the cake even faster! You can only use portals after you create both a yellow and a blue portal.

Consider the following grid:

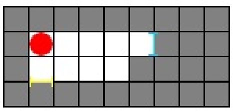


Gray cells represent walls, white cells represent empty cells, and the red circle indicates your position.

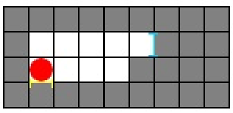
Suppose you shoot a blue portal east. The portal is created on the first wall it hits, resulting in:



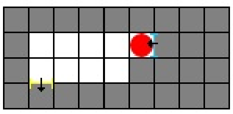
Now suppose you shoot a yellow portal south:



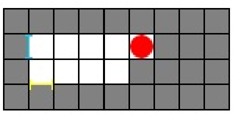
Next you move south once:



Now comes the interesting part. If you move south one more time you go through the yellow portal to the blue portal:



There can only be one yellow portal and one blue portal at any time. For example if you attempt to create a blue portal to the west the other blue portal will disappear:



A portal disappears only when another portal of the same color is fired.

Note that the portals are created on one side of the wall. If a wall has a portal on its east side you must move into the wall from the east to go through the portal. Otherwise you'll be moving into a wall, which is improbable.

Finally, you may not put two portals on top of each other. If you try to fire a portal at a side of a wall that already has a portal, the second portal will fail to form.

Given the maze, your initial position, and the cake's position, you want to find the minimum number of moves needed to reach the cake if it is possible. Remember that shooting the portal gun does not count as a move.

Input

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

The first line of each test case will contain the integers **R** and **C** separated by a space. **R** lines follow containing **C** characters each, representing the map:

- `.` indicates an empty cell;
- # indicates a wall;
- 0 indicates your starting position; and
- X indicates the cake's position.

There will be exactly one 0 and one X character per case.

Cells outside of the grid are all walls and you may use them to create portals.

Output

For each test case you should output one line containing "Case #**X**: **Y**" (quotes for clarity) where **X** is the number of the test case and **Y** is the minimum number of moves needed to reach the cake or "THE CAKE IS A LIE" (quotes for clarity) if the cake cannot be reached.

Limits

Small dataset

N = 200
1 <= **R**, **C** <= 8

Large dataset

N = 50
1 <= **R**, **C** <= 15

Sample

Input	Output

<pre>3
4 7
.0..##.
.#.....
.#.####
.#...X.
5 5
0....
.....
.....
.....
....X
1 3
0#X</pre>
	<pre>Case #1: 4
	Case #2: 2
	Case #3: THE CAKE IS A LIE</pre>

Here is the sequence of moves for the first case (note that shooting the portal gun does not count as a move):

1. Move one step east.
2. Shoot a blue portal north.
3. Shoot a yellow portal south.
4. Move one step north through the blue portal.
5. Shoot a blue portal east.
6. Move one step south through the yellow portal.
7. Move one step west.
8. Eat your delicious and moist cake.

Submissions

How Big Are the Pockets?

5pt Not attempted **311/426 users** correct (73%)

15pt Not attempted **195/249 users** correct (78%)

Portal

10pt Not attempted **159/207 users** correct (77%)

15pt Not attempted **123/163 users** correct (75%)

No Cheating

10pt Not attempted **510/637 users** correct (80%)

20pt Not attempted **68/114 users** correct (60%)

Endless Knight

5pt Not attempted **845/879 users** correct (96%)

20pt Not attempted **32/173 users** correct (18%)

Top Scores

bmerry	100
yuhch123	100
halyavin	85
wata	80
iwi	80
Ahyangyi	80
tourist	80
gawry	80
vlad89	80
neal.wu	80

Problem C. **No Cheating**

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 10 points — Solve C-small

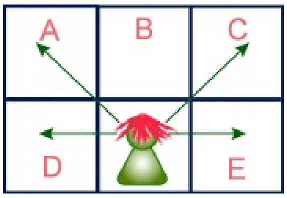
Large input 20 points — Solve C-large

Problem

A local high school is going to hold a final exam in a big classroom. However, some students in this school are always trying to see each other's answer sheet during exams!

The classroom can be regarded as a rectangle of **M** rows by **N** columns of unit squares, where each unit square represents a seat.

The school principal decided to set the following rule to prevent cheating: Assume a student is able to see his left, right, upper-left, and upper-right neighbors' answer sheets. The assignment of seats must guarantee that nobody's answer sheet can be seen by any other student.



As in this picture, it will not be a good idea to seat anyone in A, C, D, or E because the boy in the back row would be able to see their answer sheets. However, if there is a girl sitting in B, he will not be able to see her answer sheet.

Some seats in the classroom are broken, and we cannot put a student in a broken seat.

The principal asked you to answer the following question: What is the maximum number of students that can be placed in the classroom so that no one can cheat?

Input

The first line of input gives the number of cases, **C**. **C** test cases follow. Each case consists of two parts.

The first part is a single line with two integers **M** and **N**: The height and width of the rectangular classroom.

The second part will be exactly **M** lines, with exactly **N** characters in each of these lines. Each character is either a '.' (the seat is not broken) or 'x' (the seat is broken, lowercase x).

Output

For each test case, output one line containing "Case #**X**: **Y**", where **X** is the case number, starting from 1, and **Y** is the maximum possible number of students that can take the exam in the classroom.

Limits

C = 20

Small dataset

1 ≤ **M** ≤ 10
1 ≤ **N** ≤ 10

Large dataset

1 ≤ **M** ≤ 80
1 ≤ **N** ≤ 80

Sample

Input Output

4 2 3 ... ... 2 3 x.x xxx 2 3 x.x x.x 10 10 ....x..... ....x....... .... x...x.x... ....x ...x...... ....x. .x...x....	Case #1: 4 Case #2: 1 Case #3: 2 Case #4: 46

Problem D. **Endless Knight**

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 5 points	Solve D-small
Large input 20 points	Solve D-large

Problem

In the game of chess, there is a piece called the knight. A knight is special -- instead of moving in a straight line like other pieces, it jumps in an "L" shape. Specifically, a knight can jump from square (r1, c1) to (r2, c2) if and only if $(r1 - r2)^2 + (c1 - c2)^2 = 5$.

In this problem, one of our knights is going to undertake a chivalrous quest of moving from the top-left corner (the (1, 1) square) to the bottom-right corner (the (**H**, **W**) square) on a gigantic board. The chessboard is of height **H** and width **W**.

Here are some restrictions you need to know.

- The knight is so straightforward and ardent that he is only willing to move towards the right *and* the bottom. In other words, in each step he only moves to a square with a bigger row number and a bigger column number. Note that, this might mean that there is no way to achieve his goal, for example, on a 3 by 10 board.
- There are **R** squares on the chessboard that contain rocks with evil power. Your knight may not land on any of such squares, although flying over them during a jump is allowed.

Your task is to find the number of unique ways for the knight to move from the top-left corner to the bottom-right corner, under the above restrictions. It should be clear that sometimes the answer is huge. You are asked to output the remainder of the answer when divided by 10007, a prime number.

Input

Input begins with a line containing a single integer, **N**. **N** test cases follow.

The first line of each test case contains 3 integers, **H**, **W**, and **R**. The next **R** lines each contain 2 integers each, **r** and **c**, the row and column numbers of one rock. You may assume that (1, 1) and (**H**, **W**) never contain rocks and that no two rocks are at the same position.

Output

For each test case, output a single line of output, prefixed by "Case #**X**: ", where **X** is the 1-based case number, followed by a single integer indicating the number of ways of reaching the goal, modulo 10007.

Limits

$1 \le \mathbf{N} \le 100$
$0 \le \mathbf{R} \le 10$

Small dataset

$1 \le \mathbf{W} \le 100$
$1 \le \mathbf{H} \le 100$
$1 \le \mathbf{r} \le H$
$1 \le \mathbf{c} \le W$

Large dataset

$1 \le \mathbf{W} \le 10^8$
$1 \le \mathbf{H} \le 10^8$
$1 \le \mathbf{r} \le H$
$1 \le \mathbf{c} \le W$

Sample

Input	Output
5 1 1 0 4 4 1 2 1 3 3 0 7 10 2	Case #1: 1 Case #2: 2 Case #3: 0 Case #4: 5 Case #5: 1

```
1 2
7 1
4 4 1
3 2
```