Round 3 2011

A. Irregular Cakes

B. Dire Straights

C. Perpetual Motion

D. Mystery Square

Contest Analysis

Questions asked

Submissions

Irregular Cakes

7pt	Not attempted **365/378 users** correct (97%)
7pt	Not attempted **347/365 users** correct (95%)

Dire Straights

4pt	Not attempted **338/374 users** correct (90%)
12pt	Not attempted **267/315 users** correct (85%)

Perpetual Motion

5pt	Not attempted **209/218 users** correct (96%)
24pt	Not attempted **91/99 users** correct (92%)

Mystery Square

10pt	Not attempted **317/342 users** correct (93%)
31pt	Not attempted **1/46 users** correct (2%)

Top Scores

linguo	84
nika	69
winger	69
zyz915	69
misof	69
andrewzta	69
rng..58	69
mystic	69
ACRushTC	69
natalia	69

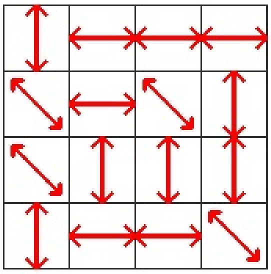
Problem C. Perpetual Motion

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 C-small
Large input 24 points	Solve C-large

Problem

Have you ever been to the Google Lemming Factory? It is a very unusual place. The floor is arranged into an **R** x **C** grid. Within each grid square, there is a conveyor belt oriented up-down, left-right, or along one of the two diagonals. The conveyor belts move either forwards or backwards along their orientations, and you can independently choose which of the two possible directions each conveyor belt should move in.

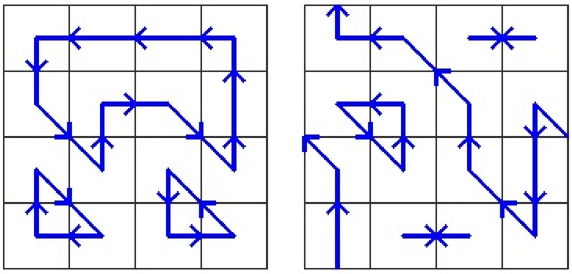


Currently, there is a single lemming standing at the center of each square. When you start the conveyor belts, each lemming will move in the direction of the conveyor belt he is on until he reaches the center of a new square. All these movements happen simultaneously and take exactly one second to complete. Afterwards, the lemmings will all be on new squares, and the process will repeat from their new positions. This continues forever, or at least until you turn off the conveyor belts.

- When a lemming enters a new square, he continues going in the direction he was already going until he reaches the center of that square. He will not be affected by the new conveyor belt until the next second starts.
- If a lemming moves off the edge of the grid, he comes back at the same position on the opposite side. For example, if he were to move diagonally up and left from the top-left square, he would arrive at the bottom-right square. By the miracle of science, this whole process still only takes 1 second.
- Lemmings never collide and can always move past each other without difficulty.

The trick is to choose directions for each conveyor belt so that the lemmings will keep moving forever without ever having two of them end up in the center of the same square at the same time. If that happened, they would be stuck together from then on, and that is not as fun for them.

Here are two ways of assigning directions to each conveyor belt from the earlier example:



In both cases, we avoid ever sending two lemmings to the center of the same square at the same time.

Given an arbitrary floor layout, calculate **N**, the number of ways to choose directions for each conveyor belt so that no two lemmings will ever end up in the center of the same square at the same time. The answer might be quite large, so please output it modulo 1000003.

Input

The first line of input gives the number of test cases, **T**. **T** test cases follow. Each begins with a line containing positive integers **R** and **C**.

This is followed by **R** lines, each containing a string of **C** characters chosen from "|-/\". Each character represents the orientation of the conveyor belt in a single square:

- '|' represents a conveyor belt that can move up or down.
- '-' represents a conveyor belt that can move left or right.
- '/' represents a conveyor belt that can move up-right or down-left.
- '\' represents a conveyor belt that can move up-left or down-right.

Output

For each test case, output one line containing "Case #x: **M**", where x is the case number (starting from 1), and **M** is the remainder when dividing **N** by 1000003.

Limits

1 ≤ **T** ≤ 25.

Small dataset

3 ≤ **R** ≤ 4.
3 ≤ **C** ≤ 4.

Large dataset

3 ≤ **R** ≤ 100.
3 ≤ **C** ≤ 100.

Sample

```
Input     Output

3         Case #1: 2
3 3       Case #2: 0
|-/       Case #3: 16
|||
--|
3 4
----
||||
\\//
4 4
|---
\-\|
\|||
|--\
```