Problem 4 - Tower (tower)

A Phoenix has been summoned onto the roof of the city's sacred tower. Whoever reaches the top of the tower, will be given a magical feather. But time is against them.

The sacred tower has N floors. Each floor i is a chessboard with $S[i]$ boxes each side and contains enemies with $L[i]$ level. Each floor has an entrance $in[i]$ and an exit $out[i]$, where the $in[i]$ represents the staircase to the previous floor and $out[i]$ represents the staircase to the next floor.

Players can move horizontally and vertically and a move to another box is equal to one step. If a player moves to a box containing a monster, they'll defeat the monster and will increase their own level by $L[i]$. The monster then will disappear from the box.

Every time the player reaches the $out[i]$ of the floor, the player goes to the upper floor in the $in[i+1]$ position.

Pay attention if the player reaches the $in[i]$ of the same floor for the second time the player goes to the lower floor at $out[i-1]$ position. If the player reaches the $out[N-1]$ the game finishes and the player found the solution, if the player reaches the $in[0]$ the player is at the beginning of the tower and nothing happens.

Starting from level 1, at the entrance of the first floor, the $in[0]$ box, reach the exit of the last floor, the $out[N-1]$ box at level major or equal to L with up to P steps, to receive the magical feather from the Phoenix.

Note: is it possible that there is more than one solution, so you don't need to find the best path, just a path that meets the requirements.

Input data

The first line of the input file contains an integer **T**, the number of test cases to solve, followed by **T** test cases, numbered from **1** to **T**.

In each test case, the first line contains three information about the tower:

- **N** the numbers of floors
- **L** the player level to reach
- **P** the maximum number of steps

The following lines will contain the information about the **N** floors and are divided as follows:

The first line contains the details of the floor:

- **i** the number of the current floor
- $S[i]$ the length of the chessboard
- $L[i]$ the level of monsters on the floor

The next $S[i]$ lines contains $S[i]$ space-separated characters each, describes the floor's chessboard:

- **I**: the entrance point
- **O**: the exit point
- **M**: the monster with $L[i]$ level

- *****: empty box

Output data

The output file must contain T lines. For each test case in the input file, the output file must contain a line with the characters:

Case #t: $MMM...M$

where t is the test case number (from 1 to T) and $MMM...M$ are a list of contiguous characters, representing the movements the player made to reach the final floor.

The movements can be:

- **U**: up
- **D**: down
- **L**: left
- **R**: right

Constraints

- $1 \leq T \leq 10$
- $1 \leq N \leq 50$
- $10 \leq L \leq 2\,000$
- $100 \leq P \leq 2\,000$
- $2 \leq S[i] \leq 90$
- $1 \leq L[i] \leq 20$
- You start the tower at level 1

Scoring

- **input 1** : $T = 1$, $N \leq 5$, $L \leq 50$ and $P \leq 200$
- **input 2** : $T = 3$, $N \leq 20$, $L \leq 70$ and $P \leq 150$
- **input 3** : $T = 6$, $N \leq 25$, $L \leq 150$ and $P \leq 700$
- **input 4** : $T = 8$, $N \leq 40$, $L \leq 700$ and $P \leq 1500$
- **input 5** : $T = 10$, $N \leq 50$, $L \leq 2000$ and $P \leq 2000$

Examples

input	output
1 3 10 50 0 4 1 I * * * M M * * * M * * * * * O 1 3 2 M * * * O * M * I 2 2 3 I * M O	Case #1: DRDDRRLLUURDDR

Explanation

In this example you have a tower with three floors. The first floor is four boxes long and there are three enemies at level one, the second floor is three boxes long and the enemies are at level two, the last floor is two boxes long and the enemy is at level three.

The solution to reach the highest floor is:

- 1st floor: D R D D R R
- 2nd floor: L L U U R D
- 3rd floor: D R

The path for a single tower must be displayed in a single line.