

2341 Illusive Chase

Tom the robocat is presented in a Robotics Exhibition for an enthusiastic audience of youngsters, placed around an $m \times n$ field. Tom which is turned off initially is placed in some arbitrary point in the field by a volunteer from the audience. At time zero of the show, Tom is turned on by a remote control. Poor Tom is shown a holographic illusion of Jerry in a short distance such that a direct path between them is either vertical or horizontal. There may be obstacles in the field, but the illusion is always placed such that in the direct path between Tom and the illusion, there would be no obstacles. Tom tries to reach Jerry, but as soon as he gets there, the illusion changes its place and the chase goes on. Let's call each chase in one direction (up, down, left, and right), a *chase trip*. Each trip starts from where the last illusion was deemed and ends where the next illusion is deemed out. After a number of chase trips, the holographic illusion no more shows up, and poor Tom wonders what to do next. At this time, he is signaled that for sure, if he returns to where he started the chase, a real Jerry is sleeping and he can catch it.

To simplify the problem, we can consider the field as a grid of squares. Some of the squares are occupied with obstacles. At any instant, Tom is in some unoccupied square of the grid and so is Jerry, such that the direct path between them is either horizontal or vertical. It's assumed that each time Tom is shown an illusion; he can reach it by moving only in one of the four directions, without bumping into an obstacle. Tom moves into an adjacent square of the grid by taking one and only one step.

The problem is that Tom's logging mechanism is a bit fuzzy, thus the number of steps he has taken in each chase trip is logged as an interval of integers, e.g. 2 to 5 steps to the left. Now is your turn to send a program to Tom's memory to help him go back. But to ease your task in this contest, your program should only count all possible places that he might have started the chase from.

Input

The first line of the input file contains a single integer t ($1 \leq t \leq 10$), the number of test cases, followed by the input data for each test case. The first line of each test case contains two integers m and n , which are the number of rows and columns of the grid respectively ($1 \leq m, n \leq 100$). Next, there are m lines, each containing n integers which are either '0' or '1', indicating whether the corresponding cell of the grid is empty (0) or occupied by an obstacle (1).

After description of the field, there is a sequence of lines, each corresponding to a chase trip of Tom (in order). Each line contains two positive integers which together specify the range of steps Tom has taken (inclusive), followed by a single upper-case character indicating the direction of the chase trip, which is one of the four cases of 'R' (for right), 'L' (for left), 'U' (for up), and 'D' (for down). (Note that these directions are relative to the field and are not directions to which Tom turns). This part of the test case is terminated by a line containing exactly two zeros.

Output

For each test case, there should be a single line, containing an integer indicating the number of cells that Tom might have started the chase from.

Sample Input

```
2
6 6
```

```
0 0 0 0 0 0
0 0 0 1 1 0
0 1 0 0 0 0
0 0 0 1 0 0
0 0 0 1 0 1
0 0 0 0 0 1
1 2 R
1 2 D
1 1 R
0 0
3 4
0 0 0 0
0 0 0 0
0 0 0 0
1 2 R
3 7 U
0 0
```

Sample Output

```
10
0
```