Inverse Exchange

Time limit:	2 seconds
Memory limit:	1024 megabytes

Inverse Exchange is a trading simulator with an $n \times m$ grid of markets. Each market has a binary indicator: 1 if its quotes need adjustment, and 0 if they are balanced. Rows are numbered from 1 to n from top to bottom, and columns are numbered from 1 to m from left to right. Cell (r, c) is in row r, column c.

To perform a rebalancing operation, choose two paths inside the grid: P from $(1, 1)$ to (n, m), moving one cell down or right at each step; and Q from $(1, m)$ to $(n, 1)$, moving one cell down or left at each step. The operation flips the indicator in every cell visited by at least one of the two paths, including the endpoints: 0 becomes 1, and 1 becomes 0. A cell visited by both paths is flipped only once.

You may perform at most $2nm$ rebalancing operations. Find a sequence of operations that minimizes the number of 1s left in the grid. You do not need to minimize the number of operations.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 1000$). The description of the test cases follows.

The first line of each test case contains two integers n and m, denoting the number of rows and columns of the grid ($2 \le n, m \le 50$).

The i-th of the following n lines contains a string of m characters '`0`' and '`1`', describing row i of the grid.

It is guaranteed that the sum of $n \cdot m$ over all test cases does not exceed 10^4.

Output

For each test case, print the number of operations k ($0 \le k \le 2nm$), followed by k pairs of strings P and Q, each of length $n + m - 2$. P must contain exactly $n - 1$ characters '`D`' (down) and $m - 1$ characters '`R`' (right). Q must contain exactly $n - 1$ characters '`D`' and $m - 1$ characters '`L`' (left).

Your operations must leave the minimum possible number of indicators equal to 1. You do not need to minimize k. If there are multiple answers, print any of them.

Example

standard input	standard output
3 3 4 0000 0000 0000 2 2 11 11 2 3 110 000	0 1 RD LD 2 RDR DLL DRR DLL

Note

In the third test case, the two operations leave a single 1, which is optimal. The shaded cells in the first and the second grid are flipped by the operations. Solid and dashed lines show P and Q, respectively.

