Flipping Frenzy

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Mr. Nežmah found a grid with n rows and m columns filled up with 0s and 1s. Due to his OCD, he wants to turn all the ones in the grid into zeros. He has k ordered pairs of numbers (a_i, b_i) he likes. The only operation he is allowed to do is the following:

- Choose a cell (r, c) either inside the first row or the first column.
- Choose one pair of numbers (a, b) he likes.
- Flip all the numbers inside the cells which are contained in the rectangle of height a and width b whose top left corner is the cell (r, c). Flipping means changing all zeroes into ones, and ones into zeroes. The rectangle must be fully contained in the grid.

Help Nežmah clear the grid!

Input

The first line contains a single integer t ($1 \le t \le 300$) — the number of test cases.

The first line of each test case contains 3 integers n, m, and k ($2 \le n, m \le 1000, 1 \le k \le 20$), the dimensions of the grid and the number of pairs (a_i, b_i).

The next n lines contain m characters each, that are all `0` or `1`, describing the grid.

The next k lines each contain two integers a_i and b_i ($1 \le a_i < n, 1 \le b_i < m$, all pairs (a_i, b_i) are distinct).

It is guaranteed that the sum of n and the sum of m over all test cases do not exceed 1000.

Output

For each test case, output the number of operations ($0 \le op \le k(n + m)$).

In the next op lines, output 3 integers r_i, x_i, and y_i ($1 \le r_i \le k$, $1 \le x_i \le n - a_{r_i} + 1$, $1 \le y_i \le m - b_{r_i} + 1$, at least one of x_i and y_i is equal to 1), meaning that you choose to flip all cells inside the rectangle with a_{r_i} rows and b_{r_i} columns with its upper left corner at (x_i, y_i).

A solution is guaranteed to exist, and if there are multiple solutions you can output any of them.

Example

standard input	standard output
5 6 6 2 000000 111110 000000 100001 011111 111110 5 5 3 5 4 5 2 10111 00101 10000 01100 1 3 2 1 2 2 1 00 00 1 1 5 5 3 10100 10010 00000 01101 00100 3 4 2 3 4 2 5 5 4 01010 01010 10101 00100 00100 2 2 2 3 3 2 3 3	5 1 1 2 2 1 2 2 2 1 2 3 1 2 4 1 6 1 1 1 1 1 2 1 4 1 2 1 3 2 1 5 2 3 1 0 10 1 1 1 1 1 2 1 2 1 2 1 1 2 1 2 2 4 1 3 1 1 3 1 2 3 1 4 3 2 1 5 3 1 4 2 1 3 4 1 2 2 4 1 3 3 1