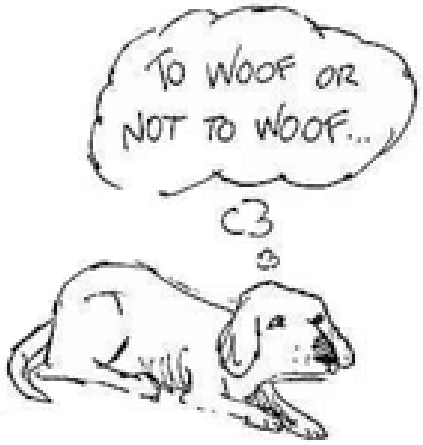
4913 Lecture Hall Seating

The world famous actor Buster is planning on performing the soliloquy "To Bark or not to Bark" at RIT. Because a large crowd is expected a lecture hall has been reserved for the event. The lecture hall contains m rows of seats, with exactly n seats per row, forming an $m \times n$ rectangle. Although the lecture hall is large, the seats are very close together. When selecting seats to sit in people only choose those seats where the seats immediately to the left, right, front and back of them are unoccupied.



Because of Buster s popularity some early birds arrived early and have already chosen their seats. Buster s contract stipulates that his fee is based on the number of real people (rather than early birds) who attend his performance. RIT needs to know the maximum number of real people who can be seated in the lecture hall, not counting the early birds, leaving adjacent seats empty and without having to reseat any of the early birds.

Input

The input file contains several test cases, each of them as described below.

The first line contains two positive integers, m and n, the number of rows and columns in the lecture hall, separated by white space (m and n are ≤ 500).

The next m lines describe the occupied positions in the respective row. Each row is represented by a sequence of zeros and ones of length n, separated by white space, where a zero corresponds to an unoccupied seat and a one corresponds to a seat occupied by an early bird. See the two examples below.

Output

For each test case, write to the output as described below.

The first line contains a positive integer p, the maximum number of real people who can be seated, not counting the unmovable early birds.

The next m lines describe a possible seating arrangement, using zeros to describe unoccupied seats, ones to describe seats with early birds, and twos to describe seats occupied by real people, all separated by single spaces.

Sample Input

```
3 5
0 0 1 0 0
0 0 0 0 0
0 1 0 0 0
3 5
1 0 0 0 0
0 0 1 0 0
0 0 0 0 0
```

Sample Output

```
4
2 0 1 0 2
0 0 0 2 0
0 1 0 0 2
4
1 0 0 2 0
0 0 1 0 2
2 0 0 2 0
```