Problem G. Projection

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

Everybody knows that you are a `TensorFlow` fan. Therefore, you've been challenged to recreate the `TensorFlow` logo from two projections.

Consider that you have a `3D` volume, $n \times m \times h$, and two projections (two matrices with dimensions $n \times m$ and $n \times h$ with elements 0 and 1). You are asked to compute a possible sets of cubes that must be placed inside the `3D` volume such that the `3D` object created with the cubes throws the shadows specified by the projection-matrices, when the light comes from left and front. If it is not possible, just print -1. If it is possible you must find exactly two sets, one with the **maximum** amount of cubes and one with the **minimum** amount. You can assume there is no gravitation (the cubes are located inside the `3D` volume exactly where they are placed, without requiring any support). We assume that 1 represents shadow and 0 represents light.

If there are multiple such solutions, you must output the minimum lexicographic one. One solution A is lexicographically smaller than another solution b if the first number that differs between the two solutions is smaller in a than in b.

For example, solution $[(0,0,0),(1,1,1)]$ is smaller than $[(1,1,1),(0,0,0)]$.

Input

The first line contains three integers separated by a single space n, m, h ($1 \leq n, m, h \leq 100$) — the volume dimensions.

Each of the next n lines contains m characters, each being either 1 or 0 representing either a shadow area (1) or a light area (0), describing the projection from the light in the front.

Each of the next n lines contains h characters, with the same format as above, describing the projection from the light on the left.

Output

The output should contain on the first line one number, either -1 if there is no solution or k_{max} representing the maximum number of cubes we can assign in the volume that will generate the two projections given in the input.

The next k_{max} lines should contain triplets of numbers x, y, z ($0 \leq x < n$, $0 \leq y < m$, $0 \leq z < h$) representing the cubes chosen in the lexicographically smallest solution with maximum number of cubes.

Then, only if there is a solution, one more line follows containing k_{min}, the minimum number of cubes we can assign in the volume that will generate the two projections given in the input.

After that, the next k_{min} lines should contain triplets of numbers x, y, z ($0 \leq x < n$, $0 \leq y < m$, $0 \leq z < h$) representing the cubes in the lexicographically smallest solution with minimum number of cubes.

Examples

standard input	standard output
5 3 3 111 010 010 010 010 111 100 110 100 100	14 0 0 0 0 0 1 0 0 2 0 1 0 0 1 1 0 1 2 0 2 0 0 2 1 0 2 2 1 1 0 2 1 0 2 1 1 3 1 0 4 1 0 8 0 0 0 0 1 1 0 2 2 1 1 0 2 1 0 2 1 1 3 1 0 4 1 0
2 2 2 00 00 11 11	-1
2 3 2 101 011 10 11	6 0 0 0 0 2 0 1 1 0 1 1 1 1 2 0 1 2 1 4 0 0 0 0 2 0 1 1 0 1 2 1

Note

A cube at coordinates (x, y, z) will generate a shadow at line x and column y in the $n \times m$ projection and line x and column z in the $n \times h$ projection (indexed from 0).