Bit inversion

SEASON 10 – SECOND ROUND

You are given **N** binary numbers, each with **M** bits (the numbers can have leading zeros). You have a special device, which can perform the following operation: for a given index of a number and a position of a bit, the device inverts all bits of the number with the given index (each 0 becomes a 1 and each 1 becomes a 0) and also it inverts the bit on the given position in all other numbers. Your goal is to use the special device in such a way that all N numbers become equal to zero.

Input (bitinversion.in)

On the first line of the input are given the numbers **N** and **M**. On each of the next N lines, a binary number with M bits is given.

Output (bitinversion.out)

If it's impossible to make all numbers equal to zero using the described operation, output the number **-1** on a single line. Otherwise, on the first line of the output print the number **K** ($0 \le K \le 1\ 000\ 000$) - the number of operations you are going to perform. On each of the following K lines print two numbers $\mathbf{x_i}$ and $\mathbf{y_i}$ ($0 \le x_i < N$, $0 \le y_i < M$, the most significant bit of each number has position M-1 and the least significant has position 0) - the index of a number and the position of a bit you'll give the device for the i-th operation. It can be proved that if it's possible to make all numbers equal to zero, you can do it with no more than 1 000 000 operations. If there are several ways to do it, output any of them.

Constraints

$1 \le N*M \le 1\ 000\ 000$

Example

Input	Output
2 3 010 100	3 0 1 0 0 1 1

Explanation

The numbers change the following way:
(010, 100) → (101, 110) → (010, 111) → (000, 000)