

# Bit inversion



СЕЗОН 10 – ВТОРИ РУНД

Дадени са ви **N** двоични числа, всяко от които с по **M** бита (могат да имат водещи нули). Разполагате със специално устройство, което може да извършва следната операция по зададени от вас номер на число и позиция на бит: обръща всички битове на зададеното число (всяка нула става единица и всяка единица става нула) и освен това обръща бита на зададената позиция във всички останали числа. Целта ви е да използвате това устройство така, че всичките ви числа да станат равни на нула.

## Вход (bitinversion.in)

На първия ред на входа се въвеждат числата **N** и **M**. На всеки от следващите **N** реда се въвежда по едно двоично число с **M** бита.

## Изход (bitinversion.out)

Ако е невъзможно всичките числа да станат равни на нула чрез прилагане на гореописаната операция, изведете на единствен ред числото **-1**. В противен случай, на първия ред от изхода изведете числото **K** ( $0 \leq K \leq 1\,000\,000$ ), равно на броя операции, които ще използвате. На всеки от следващите **K** реда изведете две числа **x<sub>i</sub>** и **y<sub>i</sub>** ( $0 \leq x_i < N$ ,  $0 \leq y_i < M$ , най-младшият бит на всяко число има индекс 0, а най-старшият има индекс **M-1**) - номера на число и позицията на бит, които задавате на устройството за **i**-тата операция. Може да се докаже, че ако е възможно да направите всички числа равни на нула, то можете да го направите с до **1 000 000** операции. Ако има повече от един начина да направите всички числа равни на нула с не повече от **1 000 000** операции, изведете който и да е от тях.

## Ограничения

$$1 \leq N * M \leq 1\,000\,000$$

## Пример

Вход	Изход
2 3	3
010	0 1
100	0 0
	1 1

## Обяснение

Числата се изменят по следния начин:

$$(010, 100) \rightarrow (101, 110) \rightarrow (010, 111) \rightarrow (000, 000)$$