Prime matrix (`matrix`)

Time limit: 1.0 seconds
Memory limit: 256 MiB

Peter has an $n \times m$ matrix of integers. In a single turn Peter can increment an element of the matrix by 1. Any element of the matrix can be chosen in a turn. A matrix is defined *prime* if and only if one of these two conditions are true:

- The matrix has a row composed of prime numbers only.
- The matrix has a column composed of prime numbers only.

Peter wants to find the minimum number of turns needed for making the the matrix *prime*. To do it, he asks for your help.

Scoring

Your program will be tested on several test cases, gathered in subtasks. To get the maximal score assigned to a subtask, your program has to solve correctly all the tests related to it.

- **Subtask 1** [**0 points**]: the example tests shown below.
- **Subtask 2** [**50 points**]: each matrix element is smaller than or equal to 1000.
- **Subtask 3** [**50 points**]: no limitations.

Input/output's format

Your program will have to read the following data from standard console input:

- Row 1: contains the integers n and m, respectively the number of rows and columns.
- the following n rows: i-th row contains the m elements that comprise the i-th row of the matrix.

Your program will have to print on the console the following:

- A single integer: the smallest number of turns to make the matrix *prime*.

Constraints

- $1 \leq n, m \leq 500$.
- Each element of the matrix is smaller than or equal to 100 000.

Examples

`stdin`	`stdout`

3 3
1 2 3
4 5 6
7 8 9
	1

2 3
4 8 8
14 2 14
	3

Explanation

- In **the first example test case** We can increment once the value 1 in position $(1, 1)$, therefore row 1 will be composed of only primes.
- In **the second example test case** We can increment three times the value 8 in position $(1, 2)$, therefore the column 2 will be composed of only primes.