B. Prime Matrix

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You've got an $n \times m$ matrix. The matrix consists of integers. In one move, you can apply a single transformation to the matrix: choose an arbitrary element of the matrix and increase it by 1. Each element can be increased an arbitrary number of times.

You are really curious about prime numbers. Let us remind you that a *prime number* is a positive integer that has exactly two distinct positive integer divisors: itself and number one. For example, numbers 2, 3, 5 are prime and numbers 1, 4, 6 are not.

A matrix is *prime* if at least one of the two following conditions fulfills:

- the matrix has a row with prime numbers only;
- the matrix has a column with prime numbers only;

Your task is to count the minimum number of moves needed to get a prime matrix from the one you've got.

Input

The first line contains two integers n, m $(1 \le n, m \le 500)$ — the number of rows and columns in the matrix, correspondingly.

Each of the following n lines contains m integers — the initial matrix. All matrix elements are positive integers. All numbers in the initial matrix do not exceed 10^5.

The numbers in the lines are separated by single spaces.

Output

Print a single integer — the minimum number of moves needed to get a prime matrix from the one you've got. If you've got a prime matrix, print 0.

Examples

input

```
3 3
1 2 3
5 6 1
4 4 1
```

output

```
1
```

input

```
2 3
4 8 8
9 2 9
```

output

```
3
```

input

```
2 2
1 3
4 2
```

output

```
0
```

Note

→ Attention

Package for this problem was not updated by the problem writer or Codeforces administration after we've upgraded the judging servers. To adjust the time limit constraint, solution execution time will be multiplied by 2. For example, if your solution works for 400 ms on judging servers, then value 800 ms will be displayed and used to determine the verdict.

Codeforces Round #166 (Div. 2)

Finished

→ Virtual participation

Virtual contest is a way to take part in past contest, as close as possible to participation on time. It is supported only ACM-ICPC mode for virtual contests. If you've seen these problems, a virtual contest is not for you - solve these problems in the archive. If you just want to solve some problem from a contest, a virtual contest is not for you - solve this problem in the archive. Never use someone else's code, read the tutorials or communicate with other person during a virtual contest.

Start virtual contest

→ Problem tags

binary search | brute force | math | number theory

No tag edit access

→ Contest materials

- Announcement
- Tutorial

In the first sample you need to increase number 1 in cell (1, 1). Thus, the first row will consist of prime numbers: 2, 2, 3.

In the second sample you need to increase number 8 in cell (1, 2) three times. Thus, the second column will consist of prime numbers: 11, 2.

In the third sample you don't have to do anything as the second column already consists of prime numbers: 3, 2.