



## Problem C

### Magic Checkerboard

Problem ID: checkerboard

Consider an  $n \times m$  checkerboard. On each cell of the checkerboard, place a positive integer. The values in each column of the checkerboard must be in strictly increasing order from top to bottom, and the values in each row of the checkerboard must be in strictly increasing order from left to right.

1	2	3	4
3	4	5	6
5	6	7	8
7	8	9	10

A Magic Checkerboard has an additional constraint. The cells that share only a corner must have numbers of different parity (Even vs Odd). Note that the following checkerboard is invalid, because 2 and 4 share only a corner and have the same parity:

1	2
4	6

The first  $4 \times 4$  example is a valid Magic Checkerboard. Given a partially filled magic checkerboard, can you fill the remaining locations on the checkerboard, so that the sum of all values is as small as possible?

### Input

Each input will consist of a single test case. Note that your program may be run multiple times on different inputs. Each input starts with a line with two space-separated integers  $n$  and  $m$  ( $1 \leq n, m \leq 2000$ ), representing the number of rows ( $n$ ) and the number of columns ( $m$ ) of the checkerboard. Each of the next  $n$  lines will contain  $m$  space-separated integers  $c$  ( $0 \leq c \leq 2000$ ), representing the contents of the checkerboard. Zero is used for cells without numbers that you must fill in. You may use any positive integers to fill in the cells without numbers, so long as you form a valid Magic Checkerboard. You are not limited to numbers  $\leq 2000$ , and the numbers are not required to be unique.

### Output

Output a single integer representing the minimum sum possible by replacing the 0 cells with positive integers to form a valid Magic Checkerboard. Output  $-1$  if it is not possible to replace the 0 cells to meet the constraints of a Magic Checkerboard.



**North American  
Invitational Programming Contest 2015**

*March 28, 2015*

Hosted by:  THE UNIVERSITY OF  
**CHICAGO**

**Sample Input 1**

```
4 4
1 2 3 0
0 0 5 6
0 0 7 8
7 0 0 10
```

**Sample Output 1**

```
88
```

**Sample Input 2**

```
4 4
1 2 3 0
0 0 5 6
0 4 7 8
7 0 0 10
```

**Sample Output 2**

```
-1
```

**Sample Input 3**

```
2 3
0 0 0
0 0 0
```

**Sample Output 3**

```
18
```