

Problem J

Grid City

To support better governmental activities, Indonesia is going to move its capital city from Jakarta to another city in Kalimantan. A new city plan has been proposed by the national city planners. The new city will be built on a rectangular-shaped land, and the proposed plan divides the land into a grid of $R \times C$ cells with R rows and C columns. The city planners thought that such grid design might represent a “modern” aspect of Indonesia.

Each cell has exactly one junction located at the center and no more than 4 streets connected to the junction. Each street is perpendicular to a horizontal or vertical line. There are only 14 possible layouts for a cell:

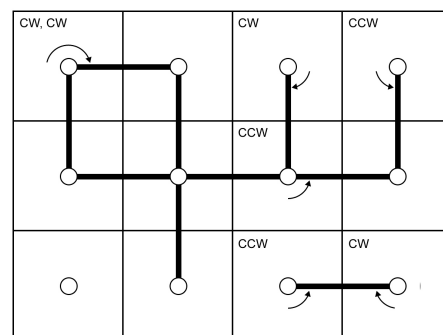
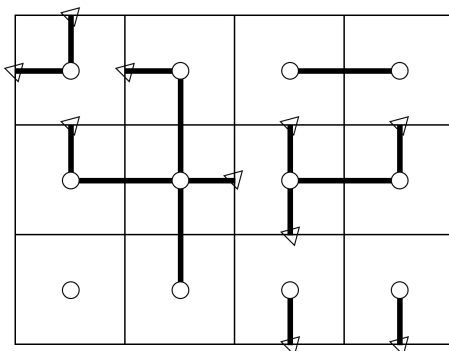
- The junction connected to exactly zero or four streets (2 possible connections).
- The junction connected to exactly two streets with an L-shaped (4 possible connections).



- The junction connected to exactly one street (4 possible connections).
- The junction connected to exactly three streets (4 possible connections).



However, after scrutinizing the plan, they found out that some streets are not ended on any junction. This is not good as those streets lose their purpose of connecting junctions. For example, consider a grid of $R = 3$ and $C = 4$ as shown in the following (left) figure. In this example, there are several streets with an end which is not on any junction (indicated by a triangle in the figure).



To remedy this problem, they can rotate one or more cells in either **clockwise** (CW) or **counter-clockwise** (CCW) direction for 90 degree. If they need to rotate a cell for 180 degree, then they are going to need 2 rotations, i.e. (CW, CW), or (CCW, CCW). The right figure shows one way such that each street properly connects two different junctions with only a total of 7 rotations.

Given a plan of $R \times C$, determine the **minimum** number of rotations needed such that every street connects two different junctions.

Input

Input begins with a line containing two integers: R C ($1 \leq R, C \leq 30$) representing the number of rows and columns in the grid, respectively. The next R lines each contains C strings representing each cell's layout. Each cell is represented by a bitstring of length 4 representing the existence of streets from its junction. Each bitstring is in the following format: $NESW$ ($N, E, S, W \in \{0, 1\}$) representing the existence of a street to the north, east, south, or west, respectively. A bit value of 1 denotes that there is a street in that respective direction, while 0 denotes that there is no such street. There will be no cell with "1010" or "0101" layout.

Output

Output in a line an integer representing the minimum number of rotations such that every street connects two different junctions, or -1 if it is impossible to do so.

Sample Input #1

```
3 4
1001 0011 0100 0001
1100 1111 1110 1001
0000 1000 0010 0010
```

Sample Output #1

```
7
```

Explanation for the sample input/output #1

This is the example from the problem description.

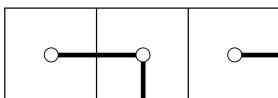
Sample Input #2

```
1 3
0100 0011 0100
```

Sample Output #2

```
-1
```

Explanation for the sample input/output #2



There is no way to rotate the cells such that every street connects two different junctions.