There’s a matrix of \( n \) rows and \( m \) columns whose elements are either 0 or 1.

If an element is 0, it is located at the \( i \)-th row and \( j \)-th column, there is at least one 1 is located at the \( i \)-th row and \( k \)-th \((k < j)\) column, and at least one 1 is located at the \( i \)-th row and \( r \)-th \((r > j)\) column, this element is called Cup.

You can swap any two rows and any two columns of the matrix. You can do this infinite times, to minimize the number of Cups.

**Input**

Multiple test cases (no more than 50), ended with EOF.

The 1st line of each test case contains two integers \( n \) and \( m \) \((3 \leq n \leq 10, 3 \leq m \leq 25)\), the number of rows and the number of columns of the matrix.

Then following \( n \) lines, each line has \( m \) space-separated 0 or 1, indicating the original configuration of the matrix.

**Output**

For each test case output a number in one line, indicating the number of minimum Cups can be obtained after the swapping operations.

**Sample Input**

```
3 3
1 1 0
1 0 1
1 0 1
0 1 1
```

**Sample Output**

```
1
```