3461 - Purifying Machine

Asia - Beijing - 2005/2006

Mike is the owner of a cheese factory. He has 2^N cheeses and each cheese is given a binary number from 00...0 to 11...1 . To keep his cheese free from viruses, he made himself a purifying machine to clean virus-infected cheese. As a talented programmer, his purifying machine is built in a special way. His purifying machine has N switches, each switch has three states, `1`, `0` and `*`. An operation of this machine is a cleaning action according to the states of the N switches. During one operation, at most one switch can be turned to state `*`, which can substitute for either `1` or `0`. When the machine is turned to a specific state, an operation will clean all the cheeses with corresponding binary numbers. For example, if N equals 6 and the switches are turned to `01*100`, the cheeses numbered `010100` and `011100` are under operation by the machine.

One day, Mike's machine was infected. When Mike found out, he had already done some operations and the cheeses operated by this infected machine were infected too. He cleaned his machine as quickly as he could, and now he needs to clean the infected cheeses with the minimum number of operations. If a cheese is infected, cleaning this cheese with the machine one or more times will make this cheese free from virus again; but if a cheese is not infected, operation on this cheese will make it go bad.

Now given the infected operations Mike has done, you need to find out the minimum number of operations that must be performed to clean all the infected cheeses without making any clean cheese go bad.

Input

There are several test cases. Each test case starts with a line containing two numbers N and M ($1 \le N \le 10$, $1 \le M \le 1000$) . N is the number of switches in the machine and M is the number of infected operations Mike has done. Each of the following M lines contains a switch state of the machine. A test case with $N = M = 0$ ends the input and should not be processed.

Output

For each test case, output one line containing an integer, which is the minimum number of operations Mike needs to do.

Sample Input

```
33
*01
100
011
00
```

Sample Output

```
2
```

Beijing 2005-2006