UVa Online Judge

1252 Twenty Questions

Consider a closed world and a set of features that are defined for all the objects in the world. Each feature can be answered with "yes" or "no". Using those features, we can identify any object from the rest of the objects in the world. In other words, each object can be represented as a fixed-length sequence of booleans. Any object is different from other objects by at least one feature.

You would like to identify an object from others. For this purpose, you can ask a series of questions to someone who knows what the object is. Every question you can ask is about one of the features. He/she immediately answers each question with "yes" or "no" correctly. You can choose the next question after you get the answer to the previous question.

You kindly pay the answerer 100 yen as a tip for each question. Because you don't have surplus money, it is necessary to minimize the number of questions in the worst case. You don't know what is the correct answer, but fortunately know all the objects in the world. Therefore, you can plan an optimal strategy before you start questioning.

The problem you have to solve is: given a set of boolean-encoded objects, minimize the maximum number of questions by which every object in the set is identifiable.

Input

The input is a sequence of multiple datasets. Each dataset begins with a line which consists of two integers, m and n: the number of features, and the number of objects, respectively. You can assume $0 < m \leq 11$ and $0 < n \leq 128$. It is followed by n lines, each of which corresponds to an object. Each line includes a binary string of length m which represent the value ("yes" or "no") of features. There are no two identical objects.

The end of the input is indicated by a line containing two zeros. There are at most 100 datasets.

Output

For each dataset, minimize the maximum number of questions by which every object is identifiable and output the result.

Sample Input

```
8 1
11010101
11 4
00111001100
01001101011
01010000011
01100110001
11 16
01000101111
01011000000
01011111001
01101101001
01110010111
01110100111
10000001010
```

```
10010001000
10010110100
10100010100
10101010110
10110100010
11001010011
11011001001
11111000111
11111011101
11 12
10000000000
01000000000
00100000000
00010000000
00001000000
00000100000
00000010000
00000001000
00000000100
00000000010
00000000001
00000000000
9 32
001000000
000100000
000010000
000001000
000000100
000000010
000000001
000000000
011000000
010100000
010010000
010001000
010000100
010000010
010000001
010000000
101000000
100100000
100010000
100001000
100000100
100000010
100000001
100000000
111000000
110100000
110010000
```

```
110001000
110000100
110000010
110000001
110000000
0 0
```

Sample Output

```
0
2
4
11
9
```