6. Subpattern Coverage

PROBLEM: Given a 2-dimensional array, find the smallest *TL subarray* that can exactly tile the original array. A *TL subarray* is part of the array that starts in the *T*op *L*eft corner of the original array.

For example, consider the 12 x 8 array below:

1	1	1	1	1	1	1	1
1	0	1	0	1	0	1	0
0	1	0	1	0	1	0	1
0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1
1	0	1	0	1	0	1	0
0	1	0	1	0	1	0	1
0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1
1	0	1	0	1	0	1	0
0	1	0	1	0	1	0	1
0	0	0	0	0	0	0	0

The smallest *TL subarray* that can exactly tile the original is the 4 x 2 array that is shaded above. When it tiles the array, it appears 12 times: 4 times on each of the 3 rows. By smallest, we refer to the number of elements in the array.

Here are some other *TL* subarrays and the reason why they don't tile the original:

1	1
1	0

This TL subarray tiles the first two rows only.

1	1	1

This TL subarray doesn't even tile the first row!

0	1	1	0
0	0	0	0

This isn't a valid TL subarray of the array above.

INPUT: There will be 10 lines of input. Each line will contain an array of 0s and1s. The format of each line is the integer *R*, followed by the integer *C*, followed by *R* hex strings that, when each hex digit is converted to 4 binary digits, will fill the *R* rows of the array from top to bottom and left to right.

If the conversion produces more digits than needed, delete the unneeded trailing digits. For example, if there are 5 columns A7 would convert to binary as 10100111 and the 5 columns would be filled with 10100.

OUTPUT: For each line of input, find the smallest *TL subarray* that exactly tiles the input array. Print the size of the *TL* subarray as two numbers: the number of rows followed by the number of columns. We guarantee that there is a unique answer to each input array. If there is no *TL* subarray smaller than the input array that can tile the input array, then print the size of the input array.

6. Subpattern Coverage

SAMPLE INPUT

```
8 8 FF AA 55 00 FF AA 55 00
4 4 F F F F
4 4 1 1 1 1
3 4 A A A
4 8 CC AA CC AA
4 6 22 B1 22 B1
6 4 3 A F 3 A F
6 6 B1 D2 21 B1 D2 21
6 8 AA AA AA AA AA AA
5 5 00 00 00 00 00
```

SAMPLE OUTPUT

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

TEST DATA

TEST INPUT

```
8 8 BC DE BC DE BC DE BC DE
4 4 A A A A
6 6 24 2C 34 24 2C 34
2 8 11 11
2 12 AAA 555
9 4 A 8 4 A 8 4 A 8 4
6 8 AF BE CD DC EB FA
5 5 01 01 01 01 01
4 1 0 0 0 0
12 8 FF AA 55 00 FF AA 55 00 FF AA 55 00
```

TEST OUTPUT

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