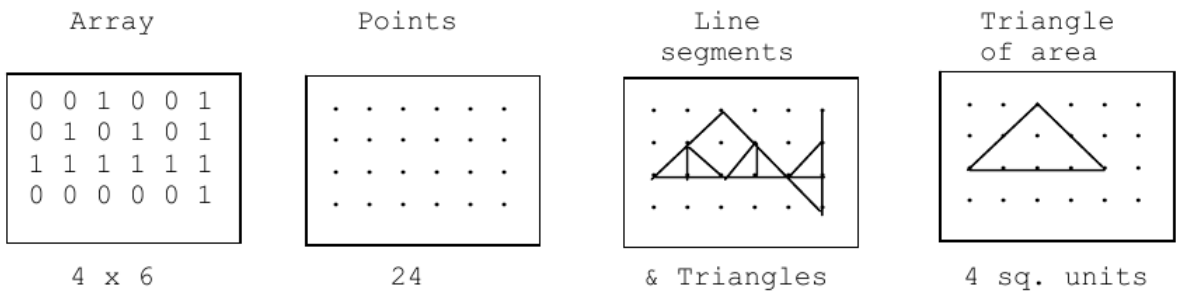
ACM-ICPC Live Archive

3258 Hidden Triangle

Given a two dimensional array of '0's and '1's, you are required to write a program that recognizes the largest (in area) right-angled isosceles triangle hidden in the array.

Assume that each '0' or '1' in the array represents a point on a plane and the distance between each pair of neighbouring points row wise or column wise is unity. Assume further that every neighbouring pair of '1's, row wise, column wise or diagonally is connected by a line segment. Two line segments emerging from a point, either join together to form a longer line segment or form an angle of 45°, 90° or 135°, thus forming right-angled isosceles triangles. The existence of hidden right-angled isosceles triangles in an array is illustrated in the figure below.



Input

Input consists of multiple test cases.

For each test case the first line gives three integers: the case number k, the number of rows m and the number of columns n of the given array. A space appears between two neighbouring integers.

Each of the next m lines gives a string of '0's and '1's of length n; the i-th line gives the i-th row of the array.

Input terminates with a value zero for case number k.

Output

For each test case, display output in one line. The line contains the case number k and the area of the largest right-angled isosceles triangle hidden in the array. The area is a real number with one digit after the decimal point. If a triangle does not exist then output '0.0' as the area.

Sample Input

```
1 3 3
101
100
101
2 4 6
001001
010101
111111
000001
0
```

Sample Output

```
1 0.0
2 4.0
```