

2517 Moving Object Recognition

There are many scenarios in which we need to know how fast an object is moving. For example, the air-traffic controller in an airport would like to know the speed of a descending plane and give warning if it's too fast or too slow. The speed detection cameras have helped the police to catch the speeding drivers, and so on.

In this problem, you are supposed to write a program to recognize the speed of a moving object.

Assume that an object is moving on a two dimensional plane, with a constant speed and direction. A camera is taking a shot every second. Assume the camera can record the real situation of the plain — you don't need to worry about the view-angle.

The background is black and the only object is white, except some noise introduced by the camera. Anyway, you can assume that the largest continuous white area is always the object. There is always only one largest continuous white area in this problem.

Given several pictures taken by the camera, calculate the speed of the object. Output the speed of the object.

A picture is represented by a matrix which contains '.' or 'x' only, where '.' in the matrix means a black block and 'x' means a white block. There are at least two pictures.

Here are some detailed definitions:

- The edge of each block is 1 mm (That is, each '.' or 'x' in the matrix represents one 1mm × 1mm block.)
- “The speed of the object” is defined by the moving speed of the center of the object as geometric center :

$$\left(\frac{\int_{(x,y) \in Object} x dx dy}{\int_{(x,y) \in Object} dx dy}, \frac{\int_{(x,y) \in Object} y dx dy}{\int_{(x,y) \in Object} dx dy} \right).$$

Anyway, all the objects are made of square blocks. The geometric center of a single square block is the center of the block. So the geometric center of the object can be calculated as:

$$\left(\frac{\sum_{i \in Object's Block} X[i]}{N}, \frac{\sum_{i \in Object's Block} Y[i]}{N} \right).$$

Where $(X[i], Y[i])$ is the coordinates of the i -th box's center and N is the number of blocks in the object.

- A white area is a set of white blocks.
- A white area is called “continuous white area” if and only if there is always a path connecting two arbitrary blocks in this area. All the blocks in the path are within this area. And each pair of sequential blocks along this path shares a common edge.
- The average speed is calculated as described in the physics textbook:

$$AvgSpeed = \frac{\sum_{t=0}^{T-1} \frac{pos(t+T) - pos(t)}{T}}{T}$$

where t is over all the discrete observation time from 0 to $T - 1$. In this problem, we define T as half of the number of observation points. (There are always even number observation points in the test suit.) $pos(t)$ is defined as the observed position of the object's center in time t , that is,

$$\left(\frac{\sum_{i \in \text{Object's Block}} X[i]}{N}, \frac{\sum_{i \in \text{Object's Block}} Y[i]}{N} \right)$$

calculated from the t -th image.

- The positive direction of X axle is from the left to the right; the positive direction of Y axle is from the top to the bottom.

Note that you may get different shape of an object in different pictures.

Input

The input is a file containing several data cases. Each case begins with a line, with only two numbers: m and k . Then follows by several $m \times k$ (m columns, k rows) matrices. Each matrix represents one picture taken by the camera. There are at least two pictures. There is always a separator line between each matrix inside a data set. The separator line is a line with m '-'s. The last matrix of a data set is followed by a terminating line — a line with m '='s.

A line with two zeros presents the end of the input file.

The picture size is at most 256×256 . There are at most 256 pictures in one test case.

Output

For each test case, output a line with two numbers which represent the speed of the object in X and Y direction respectively. The accuracy is up to 2 digits after decimal point. The speeds of the objects are in mm/s.

Sample Input

```
10 5
.....x
....xxx.x
...xxx...
....xxx..
x.....
-----
.....x
.....x
...xxx...
..xxx....
x..xxx...
=====
0 0
```

Sample Output

```
-2.00 1.00
```