

Problem L

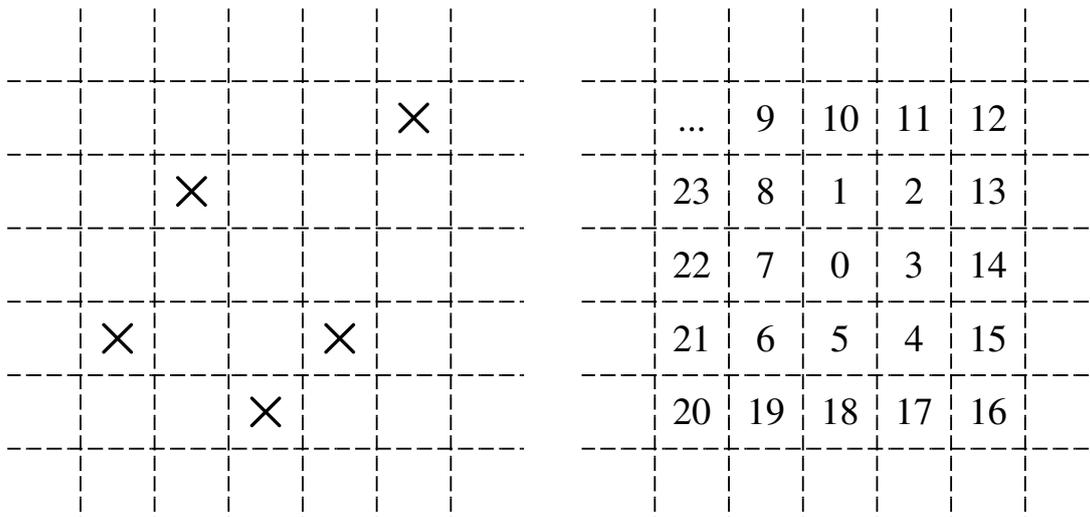
Where Am I?

Time limit: 2 seconds

Who am I? What am I? Why am I? These are all difficult questions that have kept philosophers reliably busy over the past millennia. But when it comes to “Where am I?”, then, well, modern smartphones and GPS satellites have pretty much taken the excitement out of that question.

To add insult to the injury of newly unemployed spatial philosophers formerly pondering the “where” question, the Instant Cartographic Positioning Company (ICPC) has decided to run a demonstration of just how much more powerful a GPS is compared to old-fashioned maps. Their argument is that maps are useful only if you already know where you are, but much less so if you start at an unknown location.

For this demonstration, the ICPC has created a test area that is arranged as an unbounded Cartesian grid. Most grid cells are empty, but a finite number of cells have a marker at their center (see Figure L.1(a) for an example with five marked cells). All empty grid cells look the same, and all cells with markers look the same. Suppose you are given a map of the test area (that is, the positions of all the markers), and you are placed at an (unknown to you) grid cell. How long will it take you to find out where you actually are? ICPC’s answer is clear: potentially a very, very long time, while a GPS would give you the answer instantly. But how long exactly?



(a) Grid with markers (X) corresponding to the first sample input. (b) The order in which a test subject visits grid cells, relative to starting cell 0.

Figure L.1: Sample grid and the order in which test subjects explore the grid.

In the trial, test subjects will explore their environment by following an expanding clockwise spiral whose first few steps are shown in Figure L.1(b). The starting cell is labeled “0”, and the numbers show the order in which other cells are visited. The test subjects can see a marker only if they are at its grid cell, and they will stop their exploration as soon as they know where they are based on the grid cells that they have seen so far. That means that the observed sequence of empty and marked grid cells could have begun only at a single possible starting position. The grid is unbounded, but the exploration will be finite since once a test subject has seen all markers on the grid, they’ll definitely know where they are.

Having hundreds of test subjects literally running in circles can be expensive, so the ICPC figures that writing a simulation will be cheaper and faster. Maybe you can help?



Input

The input describes a single grid. It starts with a line containing two integers d_x, d_y ($1 \leq d_x, d_y \leq 100$). The following d_y lines contain d_x characters each, and describe part of the test grid. The i^{th} character of the j^{th} line of this grid description specifies the contents of the grid cell at coordinate $(i, d_y - j + 1)$. The character is either '.' or 'X', meaning that the cell is either empty, or contains a marker, respectively.

The total number of markers in the grid will be between 1 and 100, inclusive. All grid cells outside the range described by the input are empty.

Output

In ICPC's experiment, a test subject knows they will start at some position (x, y) with $1 \leq x \leq d_x$, $1 \leq y \leq d_y$.

Output three lines. The first line should have the expected number of steps needed to identify the starting position, assuming that the starting position is chosen uniformly at random. This number needs to be exact to within an absolute error of 10^{-3} .

The second line should have the maximum number of steps necessary until one can identify the starting position.

The third line should list all starting coordinates (x, y) that require that maximum number of steps. The coordinates should be sorted by increasing y -coordinates, and then (if the y -coordinates are the same) by increasing x -coordinates.

Sample Input 1

```
5 5
....X
.X...
.....
X..X.
..X..
```

Sample Output 1

```
9.960
18
(1, 4) (4, 5)
```

Sample Input 2

```
5 1
..XX.
```

Sample Output 2

```
4.600
7
(1, 1) (5, 1)
```