

The Cube

By X Maratona de Programação IME-USP, 2006  Brazil**Timelimit: 0**

In a not too distant future, people will seek games that are increasingly dangerous in order to have fun. After ultralight and bungee-jump, games must be challenging in an intellectual manner as well. In New Zealand, a game just like this was invented; it is called "The Cube" or, as known for the Japanese name, "Sokoban".

In the game, there is a bi-dimensional labyrinth with square cells; each cell is either free or occupied by a rock. In each step, you may leave the cell you occupy and move to a neighbor free cell (up, down, left or right).

A single cell from the labyrinth contains a stack of boxes. The stack can be moved from a cell i to a cell k (for instance, $k = i + 1$), neighbor of i , if and only if you are in a cell j (in this example, $j = i - 1$), also neighbor of i , and the direction ik is the same as the direction ji (which means you are pushing the box to the next cell). A box cannot be moved in any other way (you cannot pull it, for instance). Therefore, if the box reaches an edge of the labyrinth, you will not be able to move it again. Although each time you push a box you take a step, the contrary may not always be true.

One of the empty cells is marked as the final cell; and your task is to bring the box to that final cell by a sequence of steps and push's. As the box is very heavy, you want to do it in the smallest number of push's possible.

Remember that in the real life's game, there is the possibility of you sticking yourself or even get crushed by the box, making it a lot funnier.

Input

The input may contain several instances. Each instance starts with a line containing two integers r and c ($20 \geq r, c$), representing the number of lines and columns of the labyrinth.

Following there are r lines, each containing c characters. Each character describes a cell from the labyrinth. A cell occupied by a rock is indicated by `#` and an empty cell is represented by a `.` (without quotes). Your initial position is indicated by `S`, the box position is indicated by `B` and the final position for the box is indicated by `T`.

The input ends with $r = c = 0$.

Output

For each labyrinth, print a line with the number of the instance, as shown in the example below. If it is impossible to take the box to the desired final position, print "Impossivel" (without quotes, meaning "Impossible" in Portuguese).

Otherwise, you should print two integers x and y ; x indicates the number of movements (steps you walked) and y the number of push's needed for taking the box to the final cell. If there is more than a sequence with the same minimum number of push's, the number of movements must be minimized as well. Print an empty line after each instance.

Sample Input	Sample Output

1 7 SB....T 1 7 SB..#.T 7 11 ##### #T##.....# #.#.#...#### #....B....# #.#####..# #.....S...# ##### 0 0	Instancia 1 5 5 Instancia 2 Impossivel Instancia 3 28 6
---	--