Chuck’s Challenge is a game in which a player navigates a maze consisting of various tiles in a grid. The player can encounter doors that may be unlocked using keys they acquire along the way. You must write a program to determine the minimum number of doors that must be opened to reach the exit of the maze.

**Key**

<table>
<thead>
<tr>
<th>Tile</th>
<th>Meaning</th>
<th>Passable</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;</td>
<td>Entrance</td>
<td>Yes</td>
</tr>
<tr>
<td>&lt;</td>
<td>Exit</td>
<td>Yes</td>
</tr>
<tr>
<td>a-z</td>
<td>Key</td>
<td>Yes</td>
</tr>
<tr>
<td>A-Z</td>
<td>Door</td>
<td>See rules</td>
</tr>
<tr>
<td>.</td>
<td>Ground</td>
<td>Yes</td>
</tr>
<tr>
<td>+</td>
<td>Wall</td>
<td>No</td>
</tr>
<tr>
<td>~</td>
<td>Unstable floor</td>
<td>See rules</td>
</tr>
</tbody>
</table>

**Rules**

- Tiles that are diagonal to each other are not considered to be adjacent (only up, down, left, and right).
- The player begins the maze visiting the entrance tile.
- The player may only visit passable tiles that are adjacent to the tile the player is currently visiting.
- Doors are considered impassable until a tile with a matching key (‘A’ matches ‘a’, ‘B’ matches ‘b’, etc.) has been visited.
- At first, unstable floors are considered passable; however, if the player leaves an unstable floor and visits any other type of tile, it becomes impassable.
- When an unstable floor tile becomes impassable, all adjacent unstable floor tiles become impassable.
- The external edge of the maze is an impassable wall.

**Input**
The first line of input will contain the number of test cases, $T$ ($1 \leq T \leq 50$). Each test case will begin with a line containing two integers $R \ C$ ($4 \leq R, C \leq 50$). Following that will be a maze with $R$ rows and $C$ columns. Only the characters in the key will be present in the maze.

**Output**
Each test case will have a single line of output. Print the minimum number of doors that must be opened to reach the exit or print ‘Impossible’ if the exit cannot be reached.

**Sample Input**

2
8 15
++++++++++++++
 =>$........................$
+...a...*~*~*~B+
+""""""""""""""""""""""""""""""+++
+~++..b..C~+.+
+~++A++++++~+.+
+....------c<+
+++++++++++++++++

8 6
++++++
 =>$.A<+
 ++++++
 +.a+
 +.+
 +.
 +.
 +.

**Sample Output**

2
Impossible