

Blackbeard the Pirate has stashed up to 10 treasures on a tropical island, and now he wishes to retrieve them. He is being chased by several authorities however, and so would like to retrieve his treasure as quickly as possible. Blackbeard is no fool; when he hid the treasures, he carefully drew a map of the island which contains the position of each treasure and positions of all obstacles and hostile natives that are present on the island.

Given a map of an island and the point where he comes ashore, help Blackbeard determine the least amount of time necessary for him to collect his treasure.



Input

Input consists of a number of test cases. The first line of each test case contains two integers h and w giving the height and width of the map, respectively, in miles. For simplicity, each map is divided into grid points that are a mile square. The next h lines contain w characters, each describing one square on the map. Each point on the map is one of the following:

- @ The landing point where Blackbeard comes ashore.
- ~ Water. Blackbeard cannot travel over water while on the island.
- # A large group of palm trees; these are too dense for Blackbeard to travel through.
- . Sand, which he can easily travel over.
- * A camp of angry natives. Blackbeard must stay at least one square away or risk being captured by them which will terminate his quest. Note, this is one square in any of eight directions, including diagonals.
- ! A treasure. Blackbeard is a stubborn pirate and will not leave unless he collects all of them.

Blackbeard can only travel in the four cardinal directions; that is, he cannot travel diagonally. Blackbeard travels at a nice slow pace of one mile (or square) per hour, but he sure can dig fast, because digging up a treasure incurs no time penalty whatsoever.

The maximum dimension of the map is 50 by 50. The input ends with a case where both h and w are 0. This case should not be processed.

Output

For each test case, simply print the least number of hours Blackbeard needs to collect all his treasure and return to the landing point. If it is impossible to reach all the treasures, print out '-1'.

Sample Input

```
7 7
~~~~~
~#!###~
~...#.~
~~.....~
~~~@~~
.~~~~~
...~~~.
10 10
~~~~~
~~!!!###~~
~~#...###~
~~#...*##~
~~#!..*~
~~.....~
~~.....~
~~.....~
~~.....~
~~.....~
~~.....@~
~~#!.~~~~~
~~~~~
0 0
```

Sample Output

```
10
32
```