

5054 Killbots

Killbots is a simple game of evading killer robots. Who created the robots and why they have been programmed to destroy, no one knows. All that is known is that the robots are numerous and their sole objective is to destroy everything on their way. Fortunately, their creator focused on quantity rather than quality and as a result the robots are severely lacking in intelligence. Killbots is a turn-based game, played on a rectangular *grid*. Each cell can contain one of the following:

Hero (Ⓞ) — The hero is the unfortunate soul stuck in a room with a bunch of killer robots. In the original Killbots game, the player controls the hero by moving it around the grid. But in this problem, it is assumed that the hero stays at his place and does not move at all. Instead, it can destroy any robot trying to come to its cell.

Robot (+) — Robots are mechanical thugs desiring on to crush the hero with the metallic girth. They have never heard of the “Three Laws of Robotics” and would probably crush anyone who tried to explain them. Each robot will take a single step toward the hero on every turn. If the robot is in the same row or the same column as the hero, it will step directly towards the robot. Otherwise, it will take the diagonal step that makes it as closer to the hero.

Fastbot (#) — *Fastbots* are a much speedier version of the basic robot. For each move a robot makes, they take two steps. Fortunately, their increased mobility was not paired with increased intelligence; they often only succeed in destroying themselves twice as quickly as their slower brethren.

Junkheap (*) - When two or more robot or fastbots collide, they are destroyed and the resulting debris produces a junkheap. Robots or fastbots blindly ignore any junkheaps in their path and are destroyed upon crashing into them.

The following figure is an example of how the game works. The cell E5 is where the hero stands. The cells containing junkheap are marked with ‘*’. Robots and fastbots are shown with ‘+’ and ‘#’ respectively. If we start from the grid on the left, the robots at B5 and E4 take direct steps directly towards the hero (to the right and down respectively). The latter is destroyed by the hero. The one at B8 takes an up-right step to C7. The robot at H6 takes an up-left step to G5 where it crashes into junkheap and is destroyed. The robot at F7 moves to E6 (where it will be destroyed in the next step by the fastbot initially at G8. Note that all robots and fastbots move simultaneously so at the same time F7 moves to E6, the fastbot at G8 moves to F7, without collision. After a turn that all robots and fastbots move, the fastbots move another step while the robots stay. So, the fastbot at F7 moves to E6 crashing into the robot staying there. They are both destroyed leaving junkheap at E6. The game continues until all robots and fastbots are destroyed either by the hero, or crashing into other robots or junkheap.

	A	B	C	D	E	F	G	H
1				*				
2								#
3						*		
4					+			
5		+			@	*		
6								+
7		*				+		
8		+						#

	A	B	C	D	E	F	G	H
1				*				
2								
3						*		
4						#		
5			+		@	*		
6					*			
7		*	+					
8								

The goal of this problem is count the number of robots and fastbots destroyed by the hero.

Input

There are multiple test-case in the input each specifying a game grid. Each test case contains several lines of the same length. Each line describes a row of the grid, and each character of the line is a cell of the grid. Empty cells are marked by '.', and non-empty cells are marked by the special characters described in the problem description. Af each test-case, there is a line in the input containing a single '\$' character. You may assume there is one and only one cell in the input containing '@'. Each side of the grid has at least one and at most 1000 cells.

Output

For each test case, write a single line containing the number of fastbots and robots destroyed by the hero when the game ends (after all robots and fastbots are destroyed).

Sample Input

```

...*.
.....#
.....*..
....+...
.+..@.*.
.....+
.*...+.
.+...#.
$
...
..@
.*.
.+
$

```

Sample Output

```

4
1

```