

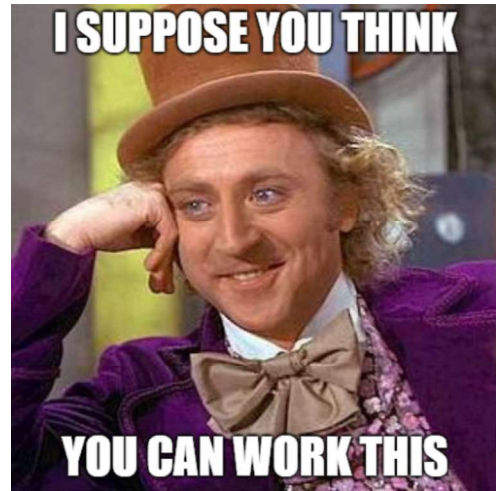
Problem.29 Wonka Factory Password

+ 22 points

The Oompa Loompas at Willy Wonka's factory have accidentally locked themselves out. Unfortunately, they've forgotten the password to the keypad used to unlock the door! Luckily, they have a password hint that they set up so they can figure out their password but since they take their security very seriously, it takes quite a bit of effort to figure out the password from the hint. Today is the day the children are coming for their tour so they don't have the time to go through the regular process of figuring out the password from the hint. This is where you come in. They've provided you with an explanation of how to get the password from some input data and are asking for you to figure out the password for them in a faster way.

Using their magical powers, the Oompa Loompas set up an infinitely long dance room. When you walk in, you see markings on the floor in a line that extends infinitely in both directions. Each spot is numbered. Directly in front of you is a spot marked 0. To the left, the spots are marked -1, -2, -3, and so on. To the right, they are marked 1, 2, 3, and so on.

The first line in your input contains a starting position telling you whether the Oompa Loompa there is standing (#) or sitting (.), starting at spot 0 and continuing towards the right. For example: ".#.#..." means at time 0 the Oompa Loompas in spots 2 and 5 are standing and the rest are sitting. The rest of your input gives you 5-character patterns (like ".##.#") you will use to pattern match. Take the initial state, then parse through the line looking for the patterns. When you find a match, it tells you the middle Oompa Loompa will be standing in the next round.



Input

The first line is "initial state: ", followed by the description of the line at time 0. The first character is the state of index 0 and the remaining characters count up. The second line holds the password at time 11. The third line holds the password at time 1000. Use these values to verify your program is running correctly. The remaining lines show the patterns that result in standing on the next round. The last line of input is a 0. Any patterns not listed result in sitting on the next round.

```
initial state: #.#.#.##.....###...###
136
19374
...##
.#..
.#...
.#.#.
.#.##
.###.
.####
#.#.#
#.###
##.#.
##.##
###..
###.#
####.
0
```

Instructions on how to process the data

Pad the current state with periods until you have ...# at the beginning of the line and #... at the end of the line. Keep track of the index number for the first # as it moves around.

```
#.#.#.##.....###...###
becomes
...#.#.#.##.....###...###...
```

Go through each pattern and search for it in the line (patterns may not exist in the line; they may also be found multiple times; they can also overlap each other). For each matched pattern, take the middle position of the pattern and store a # in that index for the next state. Below, underscores "_" denote indexes that have not been determined yet.

```
...## : .....#.#.#.##.....###...###...
new result object: _____#_____#_____

.#.. : .....#.#.#.##.....###...###...
new result object: _____#_____#_____

.#... : not found

.#.#. : .....#.#.#.##.....###...###...
new result object: _____#_____#_____

and so on...
```

