Challenge 5 - Tribblemaker

« Prev Next »

After Bones returned from planet B3/S23 to the USS Enterprise, he rushed to the bridge to brief the captain about his discovery.

– *"Jim, I found a remarkable species on the planet! They're called Tribbles. Look at their reproductive patterns!"*

In the cage, Jim and Bones could see the animals spread out on the floor on a grid. They were quickly being born and dying, based on a proximity rule.

```
---X--XX
-X-XX---
XXX---XX
XXXXX--X
X-X--XX-
--XX--XX
-X-X-X-X
X-X-X-X-
```

Spock intervened: *"Fascinating. They seem to be following the standard rules described by the earthling Dr John Conway."*

Bones added, *"Damn it, I'm a doctor not a biologist. But don't you think that* *their position on the grid seems to follow a loop? Weren't they positioned like this a while ago? I wonder if we could predict it."*

Chekov overheard the conversation and entered the room: *"I can do this! I can do this! Just let me code a quick algorithm, it will be super easy."*

Challenge

Chekov is a genius, but I'm sure you can do this as well. Given an initial grid of Tribbles, find the generations that define a loop. For example, if in generation 57 the pattern looks like it did in generation 5, we have a loop starting at generation 5 with period length 52.

The initial grid given is generation 0. The first generation you must calculate is generation 1.

Input

The file contains a grid given as an 8x8 pattern, which means it is comprised by eight lines with eight characters each. A dash (-) indicates a vacant cell and an X indicates a Tribble.

Output

Return a line containing two numbers separated by a space. The first one is the first generation of the loop and the second is the duration (period) of the loop.

The grid is guaranteed to loop eventually before reaching generation 100. The grid does NOT wrap around. Assume that any empty cells bordering the initial 8x8 cannot be populated.

First input example

Input

```
X------X
--------
---X----
---X----
---X----
--------
--------
X------X
```

Output

```
1 2
```

Second example

Input

```
XX----XX
XX----XX
--------
---XX---
---XX---
--------
XX----XX
XX----XX
```

Output

```
0 1
```

Explanation of examples

This means that for the first grid, a loop starts at generation 1 that repeats every 2 generations (i.e. the grid alternates between two states: generation 3 will look like 1, etc). For the second grid, the loop starts at generation 0 and repeats its pattern with every generation (i.e., the grid never changes).

Submit & test your code

To test and submit code we provide a set of tools to help you. Download contest tools if you haven't already done that. You will then be able to test your solution to this challenge with the challenge tokens.

```
Challenge tokens: CHALLENGE_5, CHALLENGE_SUBMIT_5
```

To test your program

```
./test_challenge CHALLENGE_5 path/program
```

A nice output will tell you if your program got the right solution or not. You can try as many times as you need.

To test your program against the input provided in the submit phase

```
./test_challenge CHALLENGE_SUBMIT_5 path/program
```

During the submit phase, in some problems, we might give your program harder inputs. As with the test token, a nice output will tell you if your program got the right solution or not. You can try as many times as you need.

In the actual contest you first need to solve the test phase before submitting the code, you must provide the source code used to solve the challenge and you can only submit once (once your solution is submitted you won't be able to amend it to fix issues or make it faster).

If you have any doubts, please check the info section.

« Prev Next »