

## Coding Exercise

A collection of particles is contained in a linear chamber. They all have the same speed, but some are headed toward the right and others are headed toward the left. These particles can pass through each other without disturbing the motion of the particles, so all the particles will leave the chamber relatively quickly.

You will be given the initial conditions by a String `init` containing at each position an 'L' for a leftward moving particle, an 'R' for a rightward moving particle, or a '.' for an empty location. `init` shows all the positions in the chamber. Initially, no location in the chamber contains two particles passing through each other.

We would like an animation of the process. At each unit of time, we want a string showing occupied locations with an 'X' and unoccupied locations with a '.'. Create a class `Animation` that contains a method `animate` that is given an `int` `speed` and a String `init` giving the initial conditions. The speed is the number of positions each particle moves in one time unit.

The method will return an array of strings in which each successive element shows the occupied locations at the next time unit. The first element of the return should show the occupied locations at the initial instant (at time = 0) in the 'X', '.' format. The last element in the return should show the empty chamber at the first time that it becomes empty.

Class: `Animation`

Method: `animate(speed, init)`, where `speed` is an integer and `init` is a String.

You may assume the following constraints:

- speed will be between 1 and 10 inclusive
- init will contain between 1 and 50 characters inclusive - each character in `init` will be '.' or 'L' or 'R'

Examples

(Note that in the examples below, the double quotes should not be considered part of the input or output strings.)

0) 2, "..R...." Returns:

```
{ "..X....",  
  "....X..",  
  ".....X",  
  "....." }
```

The single particle starts at the 3rd position, moves to the 5th, then 7th, and then out of the chamber.

1) 3, "RR..LRL" Returns:

```
{ "XX..XXX",  
  ".X.XX..",  
  "X....X",  
  "....." }
```

At time 1, there are actually 4 particles in the chamber, but two are passing through each other at the 4th position

2) 2, "LRLR.LRLR" Returns:

```
{ "XXXX.XXXX",  
  "X..X.X..X",  
  ".X.X.X.X",  
  ".X....X.",  
  "....." }
```

At time 0 there are 8 particles. At time 1, there are still 6 particles, but only 4 positions are occupied since particles are passing through each other.

3) 10, "RLRLRLRLRL" Returns:

```
{ "XXXXXXXXXX",  
  "....." }
```

These particles are moving so fast that they all exit the chamber by time 1.

4) 1, "... " Returns:

```
{ "... " }
```

5) 1, "LRRL.LR.LRR.R.LRRL." Returns:

```
{ "XXXX.XX.XXX.X.XXXX.",  
  "..XXX..X..XX.X..XX.",  
  ".X.XX.X.X..XX.XX.XX",  
  "X.X.XX...X.XXXXX..X",  
  ".X..XXX...X..XX.X..",  
  "X..X..XX.X.XX.XX.X.",  
  "..X....XX..XX..XX.X",  
  ".X.....XXXX..X..XX.",  
  "X.....X..XX...X..XX",  
  ".....X..X.XX...X..X",  
  "....X..X...XX...X..",  
  "...X..X.....XX...X.",  
  "..X..X.....XX...X",  
  ".X..X.....XX...",  
  "X..X.....XX...",  
  "..X.....XX.",  
  ".X.....XX",  
  "X.....X",  
  "....." }
```