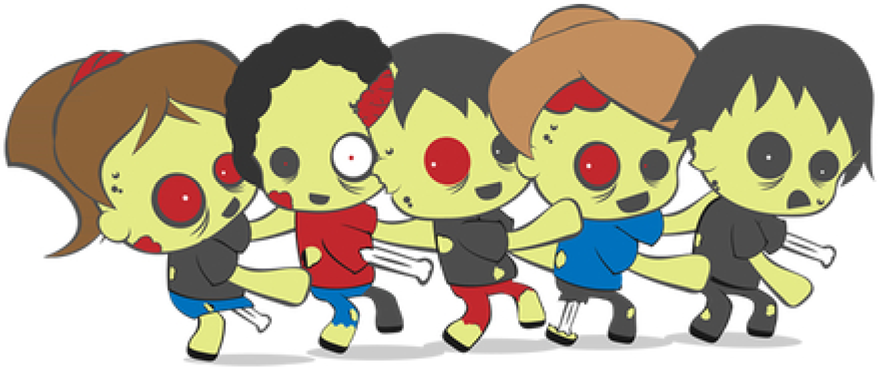
Zombie Conga Line Lab

Due Nov 29, 2020 by 11:59pm **Points** 150
Submitting a text entry box, a website url, or a file upload



Description

You are going to create a silly Zombie Conga Line using your Linked List. Each node is going to store a Zombie in it. Zombies can be Red, Yellow, Green, Blue, Magenta and Cyan. Every turn you will randomly generate a Zombie object and an action. You will then perform that action using the Zombie object as the parameter for that action.

Actions:

- Engine! - this zombie becomes the first Zombie in the conga line
- Caboose! - this zombie becomes the last zombie in the conga line
- Jump in the Line! - This zombie joins the conga line at position X where X <= length of the linked list
- Everyone Out! - Remove all matching zombies from the linked list
- You’re done! - Remove the first matching zombie from the linked list

- Brains! - Generate two more matching Zombies and add one to the front, one to the end and one to the middle.
- Rainbow Brains! - Add this zombie to the front, then add one of each zombie color to the end of the conga line.

!! Every 5 rounds remove the Head zombie and Tail zombie !!

//--

Setting up the List: Set up the initial Conga Line by running these actions:

1.Run a Rainbow Brains! Action

2.Run a random number (between 2 and 5) of Brains! actions

User Interface:

Ask the user how many rounds they want to run. Then generate that many random actions and fulfill them. If the conga line every empties completely due to an action tell the user that the Party is Over. Once the number of rounds has finished. Ask the user if they want to continue the party or end. If they choose to continue ask them for a new number of rounds to run.

//--

Output: Each round you'll output the entire Linked List. You can represent each zombie as a single character corresponding to their color (R, Y, G, B, M, C). Have fun, make silly messages, and color the output if you like! You'll show the zombie generated and the action generated. Then you'll show the outcome of the action.

Spelling it out for you:

You should create the following class:

Zombie

Your ListNode's should store Zombies.

Hints:

1. Overload the __str__ method for your Zombie and make sure your __str__ method for your LinkedList is suitable for representing the conga line easily.

//--

Sample output:

Round: 0

The Zombie Party keeps on groaning!

Size: 16 :: [R] [M] [G] [R] [R] [G] [G] [R] [M] [B] [Y] [M] [C] [G] [M] [R]

B zombie jumps in the front of the line! (ENGINE)

The conga line is now:

Size: 17 :: [B] [R] [M] [G] [R] [R] [G] [G] [R] [M] [B] [Y] [M] [C] [G] [M] [R]

****

Round: 1

The Zombie Party keeps on groaning!

Size: 17 :: [B] [R] [M] [G] [R] [R] [G] [G] [R] [M] [B] [Y] [M] [C] [G] [M] [R]

M zombie jumps in the front of the line! (ENGINE)

The conga line is now:

Size: 18 :: [M] [B] [R] [M] [G] [R] [R] [G] [G] [R] [M] [B] [Y] [M] [C] [G] [M] [R]

****

Round: 2

The Zombie Party keeps on groaning!

Size: 18 :: [M] [B] [R] [M] [G] [R] [R] [G] [G] [R] [M] [B] [Y] [M] [C] [G] [M] [R]

C zombie pulls up the rear! (CABOOSE)

The conga line is now:

Size: 19 :: [M] [B] [R] [M] [G] [R] [R] [G] [G] [R] [M] [B] [Y] [M] [C] [G] [M] [R] [C]

****

Round: 3

The Zombie Party keeps on groaning!

Size: 19 :: [M] [B] [R] [M] [G] [R] [R] [G] [G] [R] [M] [B] [Y] [M] [C] [G] [M] [R] [C]

Y zombie brought a whole party itself! (RAINBOW BRAINS!)

The conga line is now:

Size: 26 :: [Y] [M] [B] [R] [M] [G] [R] [R] [G] [G] [R] [M] [B] [Y] [M] [C] [G] [M] [R] [C] [R] [G] [B] [Y] [M] [C]

****

Round: 4

The Zombie Party keeps on groaning!

Size: 26 :: [Y] [M] [B] [R] [M] [G] [R] [R] [G] [G] [R] [M] [B] [Y] [M] [C] [G] [M] [R] [C] [R] [G] [B] [Y] [M] [C]

Y zombie brings its friends to the party (BRAINS!)

The conga line is now:

Size: 29 :: [Y] [Y] [M] [B] [R] [M] [G] [R] [R] [G] [G] [R] [M] [B] [Y] [Y] [M] [C] [G] [M] [R] [C] [R] [G] [B] [Y] [M] [C] [Y]

Deliverables Required

1. Follow the instructions on how to submit your lab.
2. Provide a screen shot of a full page view of your interpreter window that shows your Conga Line in action.

Data Structure Lab Rubric

Criteria	Ratings		Pts
Meets all Functionality Requirements This measures your ability to build the lab and fulfill all the functionality requirements specified.	**120 pts Full Marks**	**0 pts No Marks**	120 pts
Applies Proper Object-Oriented Programming Principles This measures the quality of your software design, (e.g. usage of abstraction, encapsulation, inheritance, polymorphism).	**25 pts Full Marks**	**0 pts No Marks**	25 pts
Follows Coding Conventions Following all code conventions specified.	**5 pts Full Marks**	**0 pts No Marks**	5 pts
			Total Points: 150