Lab 9: Java Structures/Sorting/RegEx UWYO COSC 2030

Introduction:

Ok so listen. You know how last week I apologized for making you do research in a language with which you were unfamiliar? This week I want you to rewrite more of your old code, as well as a RegEx checker, in that same unfamiliar language. And then I'm going to truncate the lab again. However, I will assure you that this isn't as bad as it seems, but I suppose you'll have to be the judge of that.

Lab 9:

First, go ahead and accept the assignment: <u>https://classroom.github.com/a/hllllf43</u> (too many I's imo)

Now, let's go through these functions piece by piece:

String Reversal / Parenthesis Checking - these should be very familiar to you, as this will be the third language in which you write these functions in this class. You'll never guess what's in Lab 10. (jk. unless..?) In any case, I'd suggest checking <u>some data structure documentation</u> in Java, but largely you can use your previously written logic as a template.

Heapify / Mergesort - As before, you've written these in this class. Go ahead and adjust the MAXSIZE variable as you see fit; for this lab the final submitted value doesn't matter too much to me. As long as it compiles and runs, I'll be happy.

RegEx - Though you may not have direct experience in programming with RegEx, through Ward's slides and <u>this</u> guide it should quickly start to make a lot of sense. It's just like a language, once you nail down the rules and learn how to read them it becomes legible. That main() function will evaluate 3 different RegEx patterns that you will define. The rules for these patterns will be as follows:

regExOne - Matches any string that contains the substring "galletas".

regExTwo - Matches a sentence that ends in a punctuation mark. (i.e. "Wow, this is a dumb rule!"

regExThree- Dealer's choice. Leave a comment above your function to briefly explain what the rule is and I'll evaluate it as described. Include numbers somewhere in your expression, unless you have a creative idea that doesn't involve numbers.

If you'd rather not think up your own rule, you're welcome to match the substring "b0r1ng!" at the end of a string instead.

Submission:

Push your singular Lab9.java file and your README.md to GitHub, but nothing else. This week I will take points for submitting extra files (yes, this includes a.out). Make sure you submit by April 27th at 11:59 pm.