CrsB Pre-Assessment Problems

1 Programming with Lists

Up until now, all of the programs you have had to write needed a straightforward traversal of the input data. Often, we need to write programs that combine multiple tasks on the same data. Then we have to think about how to organize the code. This organizational task is called *planning*.

For this part of the assignment, we're asking you to write two programs that involve multiple tasks. We will use these to set up upcoming lectures, so the goal is for you to think about how to do this, as much as to produce code. These will be graded on the correctness of answers that they produce.

Create a class called `Planning` and put both of the following methods in that class.

- Write a program called `rainfall` that consumes a `LinkedList<Double>` representing daily rainfall readings (double is the type for real numbers in Java). The list may contain the number -999 indicating the end of the data of interest. Produce the average of the nonnegative values in the list up to the first -999 (if it shows up). There may be negative numbers other than -999 in the list (representing faulty readings). If you cannot compute the average for whatever reason, return 1.

  For example, given a list containing (1, -2, 5, -999, 8), the program would return 3.

- Write a program called `maxTripleLength` that consumes a `LinkedList<String>` and produces the length of the longest concatenation of three consecutive elements. Assume the input contains at least three strings.

  For example, given a list containing ("a", "bb", "c", "dd"), the program would return 5 (for "bb", "c", "dd").

  You don't have to actually concatenate the strings to solve this, but if you want to, you can do this with +, as follows "go " + "goats"

Also provide an Examples class with up to four test cases for each of these two methods. We will not run either of these for thoroughness against broken implementations, but we are interested in seeing what cases you would choose to check within four test cases.