CSCI 230, Summer 2015
HW 3 – Solving a problem with a stack

Collaboration policy: the assignment is to be completed individually. You may use any online or textbook resources. **Be sure to site all resources** that you use. You may discuss the assignment with the instructor or with the TA. You may post general questions to piazza, and specific questions to the instructors privately on piazza. You may not post code publically.

Write an application program called `Balancer` that uses a stack, implemented with the `LinkedStack` class that we developed in class on July 22, to solve the following problem:

Given an arithmetic expression, print the expression and state whether or not the expression includes properly matched parentheses. The determination of validity must involve use of the stack to balance symbols as described on pages 84-85 in the text. Note: instantiate your stack to hold `Character` data (the java `Object` for the `char` data type).

Your application should include an `isItBalanced` method that verifies expressions sent as a `String` parameter and return `true` if the expression has balanced parentheses and `false` otherwise. The `isItBalanced` method will process the String parameter character by character and using a stack, determine it if it balanced or not.

The main method in your application program should call `isItBalanced` several times with different expressions and print results in a readable format.

Examples of expressions with properly matched parentheses include. Note, an empty string would also be balanced:

```
      5      (4)    (4+2)      4+(2-4)   (4+(2-4))*10
```

Examples of expressions with improperly matched parentheses include:

```
      0)      4+((3-5)      4+2)*10     (2+3
```

Note, you are only interested in the parentheses, assume that other characters are OK.

Due: Friday, July 24, 10pm. Since you will submit at least two files (`LinkedStack.java` and YB`alancer.java)`, put them in a folder named with your last name, zip/compress the folder, and upload it to OAKS by the due date/time.