Programming Assignment #1

CS 6353 Spring 2015

In this programming assignment, you will be using the Ruby programming language to parse simple math equations. The requirements for your submission are listed in sections 1 and 2 below. *Follow the requirements carefully and precisely – they will be thoroughly tested.* Section 3 provides a few example outputs.

I will grade your program by running it against a large test suite. You will be graded by the percentage of test cases you get correct.

The Due Date for this assignment is March 8th at 11:55pm.

1. Program Specification

Your program will read math formulae from standard input and output to standard output.

Spaces and tabs can both be used to separate tokens. Multiple spaces and tabs, or any combination thereof are allowed.
In the below specification, non-terminals are capitalized. Terminals are either in regular expression syntax, or single quotes if they are literal.
Here is a specification for valid inputs:

```
INPUT  ::= LINE INPUT | 'quit' '\n' | 'QUIT' '\n';
LINE   ::= EXPR '\n';                          # \n means enter is hit.
EXPR   ::= EXPR '+' TERM | EXPR '-' TERM | TERM;
TERM   ::= TERM '*' PEXPR | PEXPR;
PEXPR  ::= '(' EXPR ')' | '-' PEXPR | NUMBER;
NUMBER ::= [0-9]+
```

Your program should read in ALL my input, then make a determination if the input is well-formed or not. If **any** of the input is ill-formed, then your program should only print ERR, then a new line, and then exit.

Otherwise, your program should produce an answer for every line I input, on a line by itself.

The specification for your output should be:

```
OUTPUT: (OUTLINE)*  # meaning 0 or more OUTLINEs
OUTLINE: '-' NUM '\n' | NUM '\n'
```

2. Other Requirements

You will receive a 0 on this if any of these requirements are not met!

1. The program must be written entirely in Ruby, with no third party libraries (just ones that ship with the language).

2. You must not use any language feature to automatically evaluate the expressions. I will check your code to make sure you are doing your own parsing.
3. You must use a single input file NETID.rb as the name. For instance, for me, it would be jtv202.rb
4. The program must compile and run in the reference environment. Even if it works on your desktop, if it doesn't work in the reference environment, you will get a 0. This probably won't be a problem, but better safe than sorry!
5. You must submit your homework before 11:55pm on the due date. You will lose 15 points for each day you are late.
6. You must submit the homework through the course website, unless otherwise pre-approved by the professor.
7. You may not give or receive any help from other people on this assignment, except the professor or TA.
8. You may use references on the Internet to teach yourself Ruby.
9. Again, you may NOT use code from any other program, no matter who authored it. This includes 3rd party libraries (you may use anything that comes with Ruby by default).
10. As a warning, I generally catch people who collaborate with other students, who use web sites on the internet to beg or pay other people to write their code for them, etc. If I catch you, you will receive a 0 on the assignment and get reported to the department.

3. Test Cases

Below are four sample test cases for you, which I will use in my testing. Typically, I use anywhere from 20-40 test cases, and will definitely use these three. I strongly recommend you create your own test harness and come up with a large number of test cases to help you get the best possible grade.

For test cases, what one would type on the command line is **BLACK**, input is in **GREEN**, and output is in **BLUE**.

Case 1

```
ruby NETID.rb
(2 + 2) *      15
15 * 3
QUIT
60
45
```

Case 2

```
ruby NETID.rb
(2 + 2 * 15)
QUIT
32
```

Case 3

```
ruby NETID.rb
1+1
(2 + 2
QUIT
ERR
```

Case 4

```
ruby NETID.rb
1+1
2 + 2 + 2 + 2
ERR
```

Note that, for the third output, there is a missing right parenthesis, and for the fourth output, I have omitted “QUIT”.

4. Reference Environment

Credentials for the reference environment will be created and provided for you during the week.

5. Resources

Here are links to a bunch of different ruby tutorials:
https://www.ruby-lang.org/en/documentation/quickstart/
http://www.tutorialspoint.com/ruby/
http://rubylearning.com/satishtalim/tutorial.html
http://www.codecademy.com/en/tracks/ruby