SW483, Summer 2017
CS131 Homework Assignment #5, "The Calculator", in Lisp!

Part 1: Compiling and Running an Incomplete Calculator Program

For the first part of this assignment, download, compile, and run a simple, but incomplete Calculator program in Lisp. This program takes as input two integers from the console. Then it asks for a text character corresponding to an arithmetic operator (+, -, *, /, or %) and it prints the result of performing the arithmetic operation on the two integers. However, the program is incomplete, and only addition (+) works. When any other operation is used, the program terminates, printing nothing.

You can compile, run, and test the incomplete calculator program using the same steps as for previous assignments. There is a zip archive containing the code (Calculator.lisp).

To compile and run the calculator program:

1. Download Calculator.zip. It contains a Lisp source code file, calculator.lisp, a bash shell script, assignment_5_grader.sh, and a number of files with test cases: test_cases_addition.csv, test_cases_division.csv, test_cases_multiplication.csv, test_cases_remainder.csv, and test_cases_subtraction.csv.

2. Use the bash shell and change directories to the folder created by unpacking the .zip file. Then simply execute "clisp calculator.lisp" to run the calculator program.

3. Follow the instructions in the console to run the calculator program.

Part 2, Completing the Calculator Program:

In the second part of the assignment, you will complete the Calculator program. Your completed Calculator program will take as input two integers from the console. Then it will take a text character corresponding to an arithmetic operation ('+', '-', '*', '/', or '%') and it will print the result of performing the arithmetic operation on the two integers.

For example, if the integers 10 and 4 are entered followed by the plus-sign character ('+'), your program prints out the result of adding the two integers, which is 14. The output from your program should look like this:

```
Enter two integers.  Press return after each integer.
```

```
10
4
Enter an operation (+,-,*,/, or %).  Then press return.
+
14
```

In the above, the user input is in green, and the program output is in black.

If the integers 10 and 4 are entered followed by the division operator character, ("/"), your program prints out the quotient of an integer division, dividing the first integer by the second, which is 2. The output from your program should look like this:

```
Enter two integers.  Press return after each integer.
10
4
Enter an operation (+,-,*,/, or %).  Then press return.
/
2
```

Notice that for division, your program only needs to output the quotient, and not the remainder. Your program can perform the remainder operation if the percent-sign, "%", is entered, but it shouldn't do both operations at the same time (like your Divider program did).

Here are some notes, hints and instructions for completing the Calculator program:

1. The first part of the code Calculator.lisp takes in two integers and a single character from the user. You do not need to modify any of this code.

2. The second part of the code is a "cond" expression which acts somewhat like a "chained if-else":

    - The s-expression after "cond" has two elements:

      ((eq operation '+) (format t "~D" (+ firstInteger secondInteger)))

    - The first, "(eq operation '+)", evaluates to T when the operation entered is the plus symbol "+". The single quote character is needed for the plus symbol so that Lisp will not attempt to evaluate it, and will use the "eq" function to compare the value of "operation" to the plus symbol.

    - If the first element, "(eq operation '+)", evaluates to T, then the cond expression evaluates the second element,

"(format t "~D" (+ firstInteger secondInteger))"

which first evaluates the addition "(+ firstInteger secondInteger)" and then prints the result of the addition it to the console using "format".

3. A cond can have many such s-expressions, and it evaluates them in order until the first form in one of them evaluates to T or some other non-NIL value. The second s-expression,

   ((eq operation '-))

   Is an incomplete expression for printing the result if the user enters "-" for subtraction. You will have to complete it with a form that prints the result of subtracting y from x.

4. Notice that there is no code for the remaining three arithmetic operators (*,/, and %). Complete the code by adding three more s-expressions to the cond expression, one corresponding to each of the remaining arithmetic operators.

5. You may add the s-expressions in any order, but they depend on the user input, and as such they must come after the values of firstInteger, secondInteger, and operation are assigned.

6. The suggested way to do this is to copy and paste copies of the s-expression statement for the addition operation. Then you can modify the copies so that each of the arithmetic operations has a corresponding expression that tests for that operation and prints the result of performing that operation to the console. For each of these you only need to change the arithmetic operator character and the arithmetic function call.

   a. Use the "+" function if the "+" character is entered for the operation,
   b. Use the "-" function for "-",
   c. Use the "*" function for "*" (multiplication),
   d. Use the "floor" function for "/" (division),
   e. And Use the "rem" function for "%".

Part 3, Running the Autograder Script

Run the autograder script by executing "./assignment_5_grader.sh" from the bash shell in the zip directory. Your Lisp file must be named "calculator.lisp" with a lower-case "c" and all other letters lower-case.

The script will attempt to compile your calculator.lisp and will test your code by executing it using the clisp interpreter on a number of inputs. When it completes, the autograder script will print your final score (out of 100) to the console. Ignore any compiler warnings.