SW483, Summer 2017
CS131 Homework Assignment #3 in Lisp!

Part 1: Compiling and Running a Simple Multiplying Program

For the first part of this assignment, you will download, compile, and run a simple Lisp program that takes as input two integers from the console, and prints out the product of the integers.

You can complete the assignment either in Emacs or at the command line; you don't have to do both. Only instructions for the command line are included below. If you have trouble getting the command line to work (especially on Windows), email the course staff.

To compile and run the multiplier program:

1. Download Multiplier.zip. It contains a Lisp source code file, multiplier.lisp, a bash shell script, assignment_3_grader.sh, and a file with test cases called test_cases.csv.

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

3. Follow the instructions in the console to run the Multiplier program. It will ask you to enter two integers, pressing return after each. It should then print the product of the two integers to the console.

Part 2, Modifying the Multiplier Program to Create a Divider Program:

In this second part of the assignment, you will make some modifications to the Multiplier program to make it a Divider program. Your Divider program will take two integers from the user, but rather than multiply them, it should perform integer division with them, and output the quotient and the remainder.

For example, if the integers 10 and 3 are entered, the output from your Divider program should look like this:

```
Enter two integers.  Press return after each integer.
10
3
10 divided by 3 is
3 remainder 1
```

Notice that there is a carriage return after “is”, and that the word “remainder” is used to separate the quotient and remainder. “3 remainder 1” must appear on a line by itself.

Make the Multiplier program into a Divider program in the following steps:

1. Your Divider program will need to be defined in a file named divider.lisp. We suggest that you use the bash shell command “cp” to copy multiplier.lisp to divider.lisp. Do this by executing “cp multiplier.lisp divider.lisp” at the bash shell.

2. Modify the code in divider.lisp so that it prints the results of dividing the first number by the second instead of multiplying them. The text output from your program should be formatted exactly as it is above. For this part, no instructions are provided, only hints:

    a. Notice the way that multiplier.lisp uses the “format” function to print the text “__ times __ is __.” You will have to figure out how to use it to format the output as it appears above.

    b. The program has two symbols named “multiplier” and “multiplicand”. We recommend renaming these to “divisor” and “dividend”.

    c. To obtain integer division, you will need to replace the call to the multiplication function, “*” with a call to the “floor” function. (There is a “/” function like in Java, but it creates fractions, which is not what we want here). You’ll have to also use the “rem” function to obtain the remainder from the division.

    d. You’ll need three separate statements to print the quotient, the word “remainder” and the actual remainder.

3. Compile and run, checking for errors and correct behavior:

    a. You can try running your divider.lisp program by executing “clisp divider.lisp” from the bash shell.

Part 3, Running the Autograder Script

Run the autograder script by executing “./assignment_3_grader.sh” from the bash shell in the zip directory. Your Lisp file must be named “divider.lisp” with a lower-case “d” and all other letters lower-case.

The script will attempt to compile your divider.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.