SW483, Summer 2017
CS131 Homework Assignment #10, "List Recorder", in Lisp!

A List Recorder Program

In this assignment you will implement a program that records a sequence of 10 integers entered by the user, and prints the same sequence of integers back out to the console on a single line, separated by spaces. Your program should store the integers entered by the user by creating a Lisp list and placing the values in it. You will need to place the code that accepts the integers from the user in a loop that executes 10 times. You will need to place the code that prints out the integers in another loop that executes 10 times. So, in total, you'll need to create two separate loops.

Your starter code for this assignment will be a file called list_recorder.lisp, which takes 10 integers from the console and has a loop, but is otherwise incorrect because it does not print them back out to the console. You can compile, run, and test the incomplete program using the same steps as for previous assignments. There is a zip archive containing the code (list_recorder.lisp).

To compile and run the list recorder program:

1. Download ListRecorder.zip. It contains a Lisp source code file, list_recorder.lisp, a bash shell script, assignment_10_grader.sh, and a file with test cases: test_cases.csv.

2. Use the bash shell and change directories to the folder created by unpacking the .zip file.

3. Then simply execute "clisp list_recorder.lisp" and follow the instructions in the console to run the program.

Modifying the List Recorder Program to Make it Print:

The input and output from your program should ultimately look like this:

```
Enter ten integers separated by spaces, then press return.
10 3 0 -4 200 80 -95 87 23 -67
10 3 0 -4 200 80 -95 87 23 -67
```

Where the integers in green are the ones entered by the user, and the integers in black are the output of your program.

Program Implementation Hints:

The sample code you have been given stores the integers entered by the user into a list under the symbol "mylist". Your program will need to print the numbers in "mylist" back out to console, but without the open and close parentheses, '(' and ')'.

One way to do this is to implement another loop that prints each element of the list, and uses a form similar to the loop that takes the numbers into the list. In this case you can use the loop counter variable "i" and access the elements of the list as if they were an array using the "nth" function of lisp. For example, an expression like "(nth 3 mylist)" will evaluate to the 4th element of the list (where the indexes of the elements are 0-based and the first element is at index 0). The expression "(nth i mylist)" will evaluate to the element of the list at index i.

Another way to do this is to use recursion. You can implement a recursive function, called, say, "recursive-print-list" which takes an s-expression list of numbers as an argument. In previous recursive functions, our input was a number, and our base case was to test that our input was 0 or 1. In this case the argument is a list, and our base case can be the empty list, where the function prints nothing (you test for the empty list using the "null" function on the argument). If the argument is not the empty list, you can print the first element of the list (and a trailing space) using the "format" function, and using the "first" function (also known as "car") on the list to get the first element. Then simply recursively call recursive-print-list on the remainder of the list without the first element, which you can obtain using the "rest" function (also known as "cdr"). Then simply call recursive-print-list on mylist to print it.

Program Implementation Constraints:

The autograder script will force you to implement the program using loops or recursion by constraining you to use the setq only 6 times. The autograder will also constrain you to using format at most 4 times. This way you can print the message "Enter ten integers…" and use format from inside the body of your loop or recursive function.

Running the Autograder Script

Run the autograder script by executing "./assignment_10_grader.sh" from the bash shell in the zip directory. Your Lisp file must be named "list_recorder.lisp" with a lower-case "l" and all other letters lower-case.

The autograder script will search for any occurrences of "setq" and "format", in the file. The autograder script will give you a zero if it finds too many instances of each of these.

The script will attempt to compile your list_recorder.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.

The autograder does not run your program with a timeout, so be careful. If your program loops infinitely it will freeze the autograder script and you will need to press Ctrl-C to exit it.