SI 413 Fall 2013: Homework 4

Your name:

Due: Wednesday, September 11

Instructions: Review the course honor policy for written homeworks.

This cover sheet must be the front page of what you hand in. Fill out the left column in the table to the right after we go over each problem in class, according to the rubric below.

This rubric is also on the website, in more detail, under “Other Stuff”→“Grading Rubrics”.

Make sure all problems are submitted IN ORDER.

- 5: Solution is completely correct, concisely presented, and neatly written.
- 4: The solution is mostly correct, but one or two minor details were missed, or the presentation could be better.
- 3: The main idea is correct, but there are some significant mistakes. The presentation is somewhat sloppy or confused.
- 2: A complete effort was made, but the result is mostly incorrect. There may be some basic misunderstandings of the topic or the problem.
- 1: The beginning of an attempt was made, but the work is clearly incomplete.
- 0: Not submitted.

Comments or suggestions for the instructor:

What other students did you work with?

Citations (be specific about websites):
Many of these exercises are programming exercises, but you do not need to submit them electronically.

1 Adder

Scheme has a built-in function \texttt{add1} which adds 1 to its argument.

But what if we wanted to create a similar function like \texttt{add2} or \texttt{add20}? Write a \textbf{function that produces a function} \texttt{make-adder} that takes an argument \texttt{n} and produces a lambda expression (function) that takes one argument and adds \texttt{n} to it.

So for example \((\texttt{(make-adder 5) 10})\) produces 15.

2 To recurse or not to recurse

Given a list of 2-digit numbers, I want to know the largest digit that appears anywhere. For example:

\[
\begin{align*}
\texttt{(bigdigit (list 53 23 44 36 12))} & \quad \text{should return 6} \\
\texttt{(bigdigit (list 5 81 53))} & \quad \text{should return 8}
\end{align*}
\]

Write two versions of the \texttt{bigdigit} function: one using recursion (like the last HW), and one using the new things we have learned like \texttt{apply}, \texttt{map}, and \texttt{lambda}.

Hint: for the non-recursive version, I recommend you write a \texttt{lambda} function that takes a 2-digit number and produces the largest digit in that number. Then use \texttt{map} with this lambda function, and finally pass that whole list to an \texttt{apply} call that uses the built-in \texttt{max}.

a) Recursive version

b) Non-recursive version
3 The Doubler

a) Write a function \( \text{double } f \) which takes a 1-argument function \( f \) and produces a 1-argument function (as a lambda expression) that takes an argument \( x \) and applies \( f \) to it twice, like \( (f \ (f \ x)) \).

So for example calling \((\text{double } \sqrt{\text{)}} \ 16)\) is the same as calling \((\sqrt{\text{}} \ (\sqrt{\text{}} \ 16))\), which is 2.

b) Can we do \((\text{double } \text{double})\)? What does it do?

4 Pushups

A certain sports team scores points in varying increments (2, 3, 6, 7), and after each score, certain fans must perform a number of push-ups corresponding to the total score at that time.

Write a function \( \text{pushups } \text{points} \) that takes the points from the most recent score, and returns the total amount of push-ups that must be performed at that time. (You will need to use a global variable to hold the current score and the \text{set!} function to change it!)

For example, if we start with \( \text{pushups } 3 \), the returned value is 3. But if the next call is \( \text{pushups } 7 \), the returned value is 10, since that is the total score at that point.
5 Reverse order print

Write a function \texttt{(print-reverse L)} that takes a list \texttt{L} and prints its elements in reverse, one per line, and then returns \texttt{(void)}.

For example, calling \texttt{(print-reverse '(1 2 3))} should print

\begin{verbatim}
3
2
1
\end{verbatim}

6 Pretty-print inches

Write a function \texttt{(print-height inches)} that takes a number of inches and prints the feet and inches nicely. For example, calling \texttt{(print-height 70)} should cause the following to be printed:

\begin{verbatim}
5 feet 10 inches
\end{verbatim}

Once this works, make it nicer so that, for example \texttt{(print-height 73)} prints

\begin{verbatim}
6 feet 1 inch
\end{verbatim}

\texttt{(notice not inches)} and \texttt{(print-height 8)} just prints

\begin{verbatim}
8 inches
\end{verbatim}

and any other cases which seem sensible to you.