Today’s pair-programming partnerships are:

- Mr. Mataire and Mr. Boateng Assante
- Ms. Kyaruzi and Ms. DeWitt
- Mr. Mulhall and Mr. Schlager
- Mr. Champagne and Ms. Byrne
- Mr. Nordin and Mr. Dewey
- Ms. Jervis and Ms. Liddell

In each pair, the partner listed first should log in and manage the keyboard and screen.

Exercises

A. In sections 2.2.2 and 2.2.3 of the textbook, Friedman and Wand propose two implementations of the \texttt{Env} (environment) datatype, each of which supports all of the operations of the abstract data type specification (\texttt{empty-env}, \texttt{extend-env}, and \texttt{apply-env}). Neither of these implementations takes advantage of Scheme’s built-in “association list” procedures, which were introduced into the language for precisely the kind of keyed-lookup application in which environments might be used. Write and test an implementation of the \texttt{Env} datatype using association lists.

An association list is a list of pairs in which each pair has a “key” as its car and the corresponding “value” as its cdr. The predefined \texttt{assv} searches a given association list for a pair in which the car is identical, as tested by \texttt{eqv?}, to a given key. If it finds such a pair, it returns it; otherwise, it returns \texttt{#f}.

This is Exercise 2.5 from the textbook (page 39).

B. A \textit{signature} is a header comment for a procedure, giving the name of the procedure, the type of each of its arguments, and the type of its return value(s). For instance, in Figure 2.1 (page 38), Friedman and Wand supply the signature

\begin{center}
\texttt{apply-env : Env * Var -> SchemeVal}
\end{center}

for the \texttt{apply-env} procedure.

In Exercise 2.10 of the textbook (page 39), Friedman and Wand consider adding a third constructor, \texttt{extend-env*}, to the interface of the \texttt{Env} datatype. First, explain how this proposed constructor would differ from \texttt{extend-env}, and write a signature for its Scheme implementation.

Second, add a definition of \texttt{extend-env*} to the association-list implementation of \texttt{Env} that you developed in exercise A above.

Third, consider what changes, if any, you will need to make in the observer procedure \texttt{apply-env} in order to accommodate environments built by the new constructor.

Test your revised implementation.

C. Exercise 2.11 of the textbook (pages 39–40) describe another implementation of the revised \texttt{Env} datatype, the “ribcage” implementation. The ribcage is a list whose elements are pairs of lists. Within each pair, the car is a list of variables and the cdr is a list of the corresponding values. In implementations of programming languages, ribcages are often used to implement lexical scope. The variables in each “rib” are identifiers that
become bound simultaneously, like the parameters of a procedure or some local variables in a single declaration.

Write and test the ribcage implementation of the revised \textit{Env} datatype.