CSCI 4150 Introduction to Artificial Intelligence, Fall 2001
Assignment 1 (62 points): out Thursday August 30, due Thursday September 6

This assignment is to get you started programming in Scheme. The questions cover writing basic proce-
dures; using the mathematical procedures, the let and let* forms, and list creating/accessing procedures;
and writing mathematical recursive functions.

Notes on this assignment

• This assignment is to be turned in on paper.
• No handwritten solutions will be accepted. (You’re going to test your code on the computer anyway,
  aren’t you?)
• Turn in a computer printout of only your code. We don’t want a transcript of your Scheme session
testing all your procedures.
• You don’t have to typeset your code — a simple printout of a (reasonably formatted) ASCII file is fine.
• Indent your code properly! There will be deductions for improperly indented code.
• You may always assume (on all assignments in this class) that your procedures will be given valid
  inputs.

Questions

1. (10 points) Do Exercise 1 in Section 6.2 of “How to solve problems using Scheme” (HtSPUS). As the
   problem states, do not use conditionals (i.e. if, cond, and case). You should use a let or let* form in your solution.

2. (16 points) Do Exercise 1 of Section 9.1.4 of HtSPUS.

3. (16 points) Do Exercise 1 of Section 9.4 of HtSPUS. Note that you may only use procedures introduced
   in Section 9 for this problem.

4. (10 points) Do Exercise 2 in Section 8.1 of HtSPUS.

5. (10 points) Every irrational number can be represented as a continued fraction. For example:

\[ \sqrt{2} = 1 + \frac{1}{2 + \frac{1}{2 + \frac{1}{2 + \cdots}}} \]

Write a procedure (sqrt2approx n) that computes an approximation to \( \sqrt{2} \) using a truncated ver-
sion of the continued fraction above. For example:

<table>
<thead>
<tr>
<th>call</th>
<th>computes</th>
</tr>
</thead>
<tbody>
<tr>
<td>(sqrt2approx 0)</td>
<td>1</td>
</tr>
<tr>
<td>(sqrt2approx 1)</td>
<td>1 + \frac{1}{2}</td>
</tr>
</tbody>
</table>

Your procedure should work for any nonnegative integer \( n \). You will probably find it easiest to solve
this problem using two procedures (i.e. your sqrt2approx procedure would call your other proce-
dure).