CSCI 4150 Introduction to Artificial Intelligence, Fall 2001
Assignment 2 (75 points): out Thursday September 6, due Thursday September 13

Notes on this assignment

- This assignment is to be turned in electronically. Details will be forthcoming on the Assignment 2 information page on the course web site.
- The file you upload should only have `define` statements in it, not any calls to your procedures. It is important that you name your procedures exactly as they are named in the assignment.
- You may assume (as always) that your procedures will be given valid inputs.
- Indent your code properly, there may be some deductions for improperly indented code.

Questions

1. (10 points) Do Exercise 3 of Section 11.4 of "How to Solve Problems Using Scheme."
2. (10 points) Do Exercise 5 of Section 11.4 of "How to Solve Problems Using Scheme."
3. (10 points) Do Exercise 8 of Section 11.4 of "How to Solve Problems Using Scheme."
4. (10 points) Do Exercise 10 of Section 11.4 of "How to Solve Problems Using Scheme."
5. (10 points) One general form for continued fractions is:

$$a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cdots}}}$$

which can be written as $[a_0; a_1, a_2, a_3, \ldots]$. The square root of 2 (from Assignment 1) can be written as $[1; 2, 2, 2, \ldots]$. Often there is some repeating pattern starting with a_1, so only one period is represented. For example, the square root of 2 would be written as [1; 2]. Here are some other examples in this notation:

$$\begin{aligned} \sqrt{3} &= [1; 1, 2] \\ \sqrt{5} &= [2; 4] \\ \sqrt{7} &= [2; 1, 1, 1, 4] \end{aligned}$$

To implement continued fraction computations more generally in Scheme, we'll write procedures that given i will return the value a_i. These procedures will then be arguments to another procedure which computes an approximation to a continued fraction.

(a) Write the procedures `(sqrt2 i)` `(sqrt3 i)`, `(sqrt5 i)`, and `(sqrt7 i)` which compute the appropriate value of a_i. For example:

```
(sqrt2 0) ==> 1        (sqrt3 0) ==> 1        (sqrt3 3) ==> 1
(sqrt2 1) ==> 2        (sqrt3 1) ==> 1        (sqrt3 4) ==> 2
(sqrt2 2) ==> 2        (sqrt3 2) ==> 2
```

(b) Write the procedure `(cfrac-approx a-func n)` where `a-func` is a procedure of one argument (such as the procedures from part (a)) and `n` is the expansion limit (i.e. only a_0 through a_n are used in the continued fraction expansion. For example `(cf-approx sqrt2 3)` should return the same value as `(sqrtapprox 3)` from Assignment 1.

(c) The continued fraction expansion for the constant e has a simple pattern, although it is not periodic like the square root expansions:

$$e = [2; 1, 2, 1, 1, 4, 1, 1, 6, 1, 1, 8, 1, 1, 10, 1, \ldots]$$

Write the procedure `(e-cf i)` which can be used with your `cfrac-approx` procedure above to compute approximations to e.

6. (10 points) Do Exercise 2 of Section 12.4 of "How to Solve Problems Using Scheme."

7. (10 points) Write a non-recursive procedure `(mpositions lst e)` using map and/or apply which returns a list of numbers corresponding to the position of every occurrence of element `e` in the list `lst`. For example:

```
(m-positions '(1 3 5 3 3 7 2) 3) ==> (1 3 4)
```

The order of the elements in the returned list is not important. You may use the `one-to-n` procedure from Question 1.

⋆8. (5 points) Write a procedure `(permute x)` which takes a list `x` and returns a list of all permutations of the elements of `x`. For example:

```
(permute '(a b c))   ==>   ((a b c) (a c b) (b a c) (b c a) (c a b) (c b a))
```

The order of the permutations is not important.