CS61A – Homework 5.1	**Kurt Meinz**
University of California, Berkeley	Summer 2002

Topic: Mutable data, queues, tables

Lectures: Monday 7/22, Tuesday 7/23

Reading: Abelson & Sussman, Section 3.3.1–3

(If you are a hardware type you might enjoy reading 3.3.4 even though it isn't required.)

Homework due 10 AM Monday, 7/29:

Abelson & Sussman, exercises 3.16, 3.17, 3.21, 3.25, 3.27

You don't need to draw the environment diagram for exercise 3.27. Instead, use a trace to provide the requested explanations. Treat the table procedures `lookup` and `insert!` as primitive; i.e. don't trace the procedures they call. Also, assume that those procedures work in constant time. We're interested to know about the number of times `memo-fib` is invoked.

Extra for experts:

1. Abelson and Sussman, exercises 3.19 and 3.23.

Exercise 3.19 is incredibly hard but if you get it, you'll feel great about yourself. You'll need to look at some of the other exercises you skipped in this section.

Exercise 3.23 isn't quite so hard, but be careful about the O(1)—i.e. *constant*—time requirement.

2. Write the procedure `cxr-name`. Its argument will be a function made by composing `car`s and `cdr`s. It should return the appropriate name for that function:

```
> (cxr-name (lambda (x) (cadr (cddar (cadar x)))))
CADDDAADAR
```

Unix feature of the assignment: `alias`, `unalias`

Emacs feature of the assignment: `C-x 2; C-x 3; C-x 1; C-x o`