6.001 SICP Spring 2004 4/12/2004

Tutorial 9

Upcoming Deadlines

- Office hours this week: Tuesday 4/13, 3-5pm in 34-501
- Quiz 2 Tuesday, 4/13 7:30pm-9:30pm
- Lecture 18 due Friday, 4/16 at 9am
- No tutorial Monday/Tuesday next week (institute holiday)

Quiz II Review

```
(define (memoize proc)
  (let ((return '()))
    (lambda (arg)
      (if (assq arg return)
          (cadr (assq arg return))
        (let ((new-val (proc arg)))
          (set! return (cons (list arg new-val) return))
          new-val)))))
```

Exercise 1

```
(define memo-sq EXERCISE1)
```

Exercise 2

Draw the environment diagram for the following:

```
(memo-sq 5)
```

Exercise 3

Do the same for fibonacci:

```
(define memo-fib EXERCISE3)

(memo-fib 3)
```

Exercise 4

What does the following do?

```
(define (mystery x)
  (define (loop x y)
    (if (null? x)
        y
      (let ((temp (cdr x)))
        (set-cdr! x y)
        (loop temp x))))
  (loop x '()))
```

Exercises 5-9: Trie Data Structure

```
(define (make-node value) EXERCISE5)

(define (trie-node? x) EXERCISE6)

(define (node-value node) EXERCISE7)

(define (node-child item node) EXERCISE8)

(define (trie-lookup key node)
  (if (null? key)
      (node-value node)
      (let ((child (node-child (car key) node)))
        (if child
            (lookup (cdr key) child)
            #f))))

(define (trie-insert! key value node)
  EXERCISE9)
```