Adam Gincel

CS496

Homework 4

I pledge my honor that I have abided by the Stevens Honor System.

Exercise 1:

1) (no output)
2) 3
3) 4
4) 5
5) 5
   5
6) 5

Exercise 2:

1) 2
2) When defining y, x is dereferenced and its value is stored; y is not a reference to x.

Exercise 3:

1) ‘(1 2)
2) Just like before, when defining v, we dereference u and set its value to v. v is not a reference to u.

Exercise 4:

1) 1
2) 2
3) 7
4) #f

Exercise 5:

```
(define stack
  (let ((stk '()))
    (lambda (message)
      (case message
        ((empty?) (lambda () (null? stk)))
        ((push!) (lambda (x)
                   (set! stk (append (list x) stk))))
        ((pop!) (lambda ()
                  (if (null? stk)
                      (error "stack: Can't pop empty stack.")
                      (set! stk (cdr stk)))))
        ((top) (lambda ()
                 (if (null? stk)
                     (error "stack: No top of empty stack.")
                     (car stk))))
        (else (error "stack: Invalid message" message))))))
```

Exercise 6:

```
(define (ex1 v1 v2)
  (let ((f
         (let ((rList '()))
           (lambda (x)
             (set! rList (append (list x) rList))
             rList
             )
           ))
        ))
    (begin (f v1) (f v2))))
```

Exercise 7:

1) (mcons 0 (mcons 1 5))
2) (mcons 0 (mcons 1 5))

Exercise 8:

1) ‘a
2) ‘b
3) List l’s car is itself backwards; you can call mcdr unlimited times and you will just keep flipping the list around.

Exercise 9:

1) 2
2) 5
3) Parameters are passed by value, not reference. Using set! On a passed argument will not change the original’s value.

Exercise 10:

1) (mcons 1 5)
2) Parameters of type mutable pair seem to be passed by reference, as changes made to them with (set-mcdr!) are preserved after a method is invoked.