CS61A Notes – Week 6b: Midterm 2 Review Solutions

QUESTION 1. (What will Scheme print?)

What will Scheme print? If it will cause an error, simply write ERROR.

(a)

> (equal? ((lambda (x) (x x x)) 7) '(7 7 7))

ERROR

(b)

> (define x (cons 1 'x))
> (define y x)
> (set! x 1)
> y

(1 . x)

QUESTION 2. (Box-'n'-pointers)

Draw a box-and-pointer diagram for the following (the number of pairs in your final answer MATTERS). Also, fill in any blanks with the return value.

> (define a (list (list 3) 5))
> (define b (append a a))
> (set-car! (cdr b) (caddr b))
> (set-car! a (cons 3 4))
> a

((3 . 4) 5)
> b

((3) (3) (3 . 4) 5)

QUESTION 3. What are all the possible values of x after running the following Scheme code? If there can be deadlock, write DEADLOCK.

> (define x 8)
> (parallel-execute (lambda () (set! x (+ x 1)))
> (lambda () (set! x (if (even? x)
> (set! x (+ x 5))
> (+ x 50)))))

9, okay, 59, ERROR, 14
QUESTION 4.
(a) \(x\), because \(\texttt{'}x\texttt{'}\) is a quoted expression.
(b) ERROR, because \(x\) itself has not been defined yet.
(c) A compound procedure called \texttt{quote} that takes in one argument called \(x\).
(d) Again, \(x\), because it is a quoted expression. This expression is caught by the \texttt{quoted?} clause before the \texttt{application?} clause. The definition of a procedure called \texttt{quote} can never actually be used. ☐

QUESTION 5. Draw an environment diagram for the following Scheme code. Also, fill in any blanks with the return value.

```scheme
> (define foo
  (let ((x 3))
    (lambda ()
      (if (= x 1)
        x
        (* x (begin (set! x (- x 1)) (foo)))))))
> (foo)
6
> (foo)
1
```