

CS 61A Summer 2010 Week 4A Lab
Monday 7/19 Afternoon

1. Do this exercise on paper:

Draw the environmental diagram for the following expressions and use it to determine the return value of the final `(change 92)` call.

```
(define secret 42)

(define change
  (let ((fn (let ((secret 23))
              (lambda (x) (set! secret x))))
        ((x 12)))
    (lambda (secret)
      (fn secret))))

(change 92)
```

2. Exercise 3.12 of Abelson and Sussman.

3. Suppose that the following definitions have been provided.

```
(define x (cons 1 3))
(define y 2)
```

A CS 61A student, intending to change the value of `x` to a pair with `car` equal to 1 and `cdr` equal to 2, types the expression `(set! (cdr x) y)` instead of `(set-cdr! x y)` and gets an error. Explain why.

4a. Provide the arguments for the two `set-cdr!` operations in the blanks below to produce the indicated effect on `list1` and `list2`. Do not create any new pairs; just rearrange the pointers to the existing ones.

```
> (define list1 (list (list 'a) 'b))
list1
> (define list2 (list (list 'x) 'y))
list2
> (set-cdr! _____ )
okay
> (set-cdr! _____ )
okay
> list1
((a x b) b)
> list2
((x b) y)
```

4b. After filling in the blanks in the code above and producing the specified effect on `list1` and `list2`, draw a box-and-pointer diagram that explains the effect of evaluating the expression `(set-car! (cdr list1) (cadr list2))`.