**; CS326 HW2 - Nick Alvarez - Spring 2020**

**; Question 1**

(define (subst x y L)

(cond

((null? L) L)

((equal? x (car L)) (cons y (subst x y (cdr L))))

(else (cons (car L) (subst x y (cdr L))))))

**; Question 2**

(define (all-different? L)

(define (lookup x y)

(cond

((null? y) #t)

((equal? x (car y)) #f)

(else (lookup x (cdr y)))))

(cond

((null? L) #t)

((equal? #t (lookup (car L) (cdr L))) (all-different? (cdr L)))

(else #f)))

**; Question 3**

**; Tree definition**

(define T

'(13 (5 (1 () ()) (8 () (9 () ()))) (22 (17 () ()) (25 () ()) )))

**; left() definition**

(define (left T)

(cond

(else (car (cdr T)))))

**; right() definition**

(define (right T)

(cond

((null? T) #t)

(else (car (cdr (cdr T))))))

; val() definition

(define (val T)

(cond

((null? T) #t)

((car T))))

**; Question 3a**

(define (n-nodes T)

(cond

((null? T) 0)

(else (+ 1 (n-nodes (left T)) (n-nodes (right T))))))

**; Question 3b**

(define (n-leaves T)

(cond

((null? T) 0)

((and (null? (left T)) (null? (right T))) 1)

(else (+ (n-leaves (left T)) (n-leaves (right T))))))

**; Question 3c**

(define (height T)

(cond

((null? T) 0)

(else (+ 1 (max (height (left T)) (height (right T)))))))

**; Question 3d**

(define (postorder T)

(cond

((null? T) '())

(else (append (postorder (left T)) (postorder(right T)) (list (val T))))))

**; Question 4**

(define (flatten L)

(cond

((null? L) '())

((list? (car L)) (append (flatten (car L)) (flatten (cdr L))))

(else (cons (car L) (flatten (cdr L))))))