(define leaf

(lambda (x) x))

(define interior-node

(lambda (num bst1 bst2)

(list num bst1 bst2)))

(define empty-list?

(lambda (x)

(null? x)))

;;left-son

(define lson

(lambda (binTree)

(car (cdr binTree))))

;;right-son

(define rson

(lambda (bst)

(car (cdr (cdr bst)))))

;;contents-of only gets the symbol

(define contents-of

(lambda (bst)

(car bst)))

(define testTree

(interior-node

14

(interior-node

7

'()

(interior-node 12 '() '()))

(interior-node

26

(interior-node

20

(interior-node 17 '() '())

'())

(interior-node 31 '() '()))))

(define path

(lambda (number bst)

(cond

((eqv? number (contents-of bst)) '())

((> number (contents-of bst)) (cons "right" (path number (rson bst))))

(else (cons "left" (path number (lson bst)))))))

(path 17 testTree)