PRACTICAL EXERCISES

Practical Exercise 1

Write the function ***merge*** which consumes two sorted lists of integers in decreasing order *L1* and *L2* and produces the sorted list which contains all the elements in *L1* and *L2* in decreasing order. You may assume there are no duplicates in the list that is produced.

Practical Exercise 2

Write function ***singletons****, which* consumes a bst and produces the number of nodes in the tree that have exactly one child.

Practical Exercise 3

A leaf-labelled tree (llt) is one of the following:

- empty
- (*cons l1 l2*), where *l1* is a non-empty llt and *l2* is a llt
- (*cons v l*),where *v* is an integer and *l* is a llt

Draw the tree representation for the following llt:

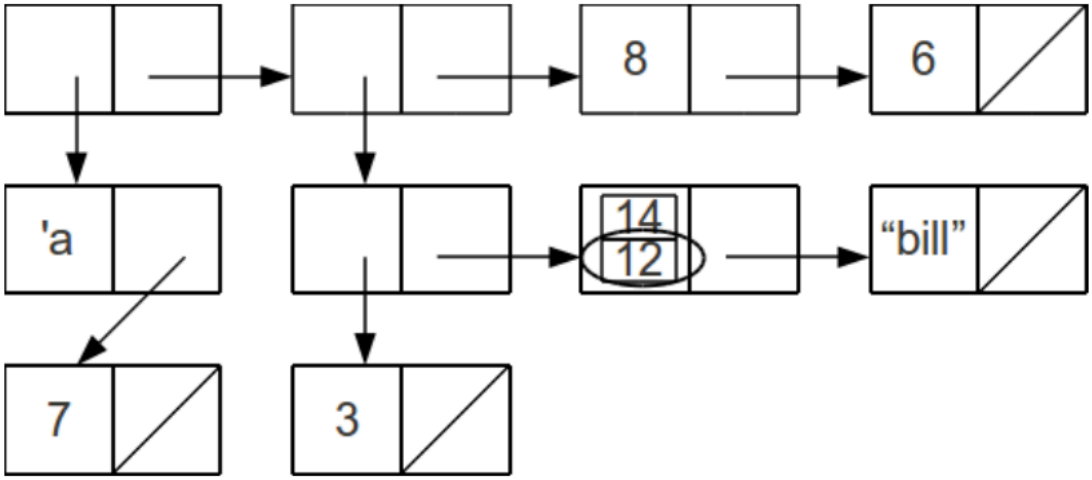
(list (1 7 (3)) 4 (5) ((8 6) 9)).

Practical Exercise 4

Write the function ***sum-leaves*** that consumes an llt *tree* and produces the sum of all the values in *tree*. The sum of the leaves in an empty tree is 0. You may not write your own *flatten* function.

Practical Exercise 5

Consider the following picture describing a list *L*:



You should notice there is a *posn* structure buried in this list (which has an *x* value of 14). Circle the correct answer for the following:

The list *L* can be expressed in *cons* notation as:

a. (cons (cons 'a (cons 7 empty))
   (cons (cons 3 empty) (cons 14 (cons 12 empty)) "bill")
   (cons 8 (cons 6 empty)))

b. (cons (cons 'a (cons 7 empty)
   (cons (cons 3 empty) (cons 14 (cons 12 empty)) "bill")
   (cons 8 (cons 6 empty)))

c. (cons (cons 'a (cons 7 empty))
   (cons (cons 3 empty) (cons (make-posn 14 12) "bill" empty))
   (cons (cons 8 empty) 6))

d. (cons 'a (cons 7 empty)
   (cons 3 (cons (make-posn 14 12) (cons "bill" empty)))
   (cons 8 (cons 6 empty)))

e. (cons (cons 'a (cons 7 empty))
   (cons (cons (cons 3 empty)
   (cons (make-posn 14 12) (cons "bill" empty)))
   (cons 8 (cons 6 empty))))