CT331 Assignment 2

Functional Programming with Scheme

Due Date: TBC

Introduction

Please submit a pdf file clearly showing the following:

1. The assignment name (“CT331 Assignment 2”)
2. Your name and student ID
3. Each question title (“Question 1”), in order, displayed clearly above your answer.
4. For each question:
    a. A copy of your code (not a screenshot) and output.
    b. Any comment you may have, or whatever textual answer the question requires.

Dr Racket is available to download from https://racket-lang.org/

Question 1

(A) In assignment_q1.rkt, write Scheme code that creates the following: [3 marks]

- A cons pair of two numbers
- A list of 3 numbers, using only the cons function.
- A list containing a string, a number and a nested list of three numbers, using only the cons function.
- A list containing a string, a number and a nested list of three numbers, using only the list function.
- A list containing a string, a number and a nested list of three numbers, using only the append function.

(B) Comment on result. Mention the difference between the cons, list and append functions. [3 marks]

Question 2

In assignment_q2.rkt, write Scheme code to do the following: [2 marks each]

A. Write a Scheme function named ins_beg which, when passed an element and a list, inserts the element at the beginning of the list. You can assume only valid elements and lists.
   e.g., if the function is called ins_beg:
   a. (ins_beg 'a '(b c d)) returns the list (a b c d)
   b. (ins_beg '(a b) '(b c d)) returns the list ((a b) b c d)

B. Write a Scheme function named ins_end which, when passed an element and a list, inserts the element at the end of the list. You can assume only valid elements and lists.
   e.g., if the function is called ins_end:
   a. (ins_end 'a '(b c d)) returns the list (b c d a)
   b. (ins_end '(a b) '(b c d)) returns the list (b c d (a b))

C. Write a Scheme function named cout_top_level that counts the number of top-levels items in a list, i.e., you don't need to include items in a sub list.

D. Write a non-tail recursive function called count_instances that counts the number of times an item occurs in a list of items (you may assume all items are atomic).

E. Write a tail-recursive version of your solution to part D called count_instances_tr

F. Write a Scheme function named count_instances_deep that counts the number of times an item occurs in a list of items; note that the list may contain sub-lists and you should also count occurrences in those lists.

Question 3

In assignment_q3.rkt, write Scheme code to do the following: [2 marks each]

A. Display in sorted order the contents of a binary search tree
B. Return #t or #f if a given item is present or absent in a tree or not. The function should take the item and a list representing a tree.
C. Insert an item correctly into a list representing a binary search tree. Your function should take an item and a tree as inputs.
D. Take a list of items and insert them into a binary search tree.
E. Implement a tree-sort algorithm. Your function should take a list of items and display them in sorted order.
F. Implement a higher order version of the tree-sort function that takes a list and a function that determines the sorted order. For example, write a version that sorts the list in ascending, descending and ascending based on last digit.