TPL Assignment

Problem Set (Ch#15)--- Q9,10

Programming Assignment – 1, 2, 4, 5, 6, 7

Problem 1. Define a Scheme function flatten that flattens a given list. The function should work with general lists potentially having sublists as elements. For example, (flatten '(A B (C (D D) C) B A)) evaluates to (A B C D D C B A).

Problem 2. Define a Scheme function slice to extract an (i; j)-slice from a given list, where i and j are two indices in the list. An (i; j)-slice of a list l is the list containing all the elements starting from the i'th element up to but not including the k'th element of l.
Note that indexing should start from 0. The function slice should behave appropriately on unexpected values for its arguments. For example, (slice 2 4 '(A B C D E)) evaluates to (C D).

Problem 3. Define a Scheme function lsort that, given a list l of lists, sorts the elements of l according to their length in ascending order; i.e. it produces a list in which shorter lists appear before longer lists in the result. Note that the order in which lists of the same length appear is not speci_ed.
For example, (lsort '((A B C) (D E) (F G H) (D E) (I J K L) (M N) (O))) evaluates to ((O) (D E) (D E) (M N) (A B C) (F G H) (I J K L)).

Problem 4. Define a Scheme function gcd that computes the greatest common divisor for two positive integers given as arguments. For example, (gcd 52 108) evaluates to 2.

Problem 5. Define a Scheme function prime-factors that constructs a list containing the prime factors in ascending order of a given positive integer. For example, (prime-factors 315) evaluates to (3 3 5 7).

Problem 6. Define a function to compute the length of a list.

Problem 7. Define a function to compute sum of squares of number of the list.