L1: Recursive programming in Lisp (1)

Write recursive Lisp functions for the following problems:

1.
a) Write a function to return the n-th element of a list, or NIL if such an element does not exist.
b) Write a function to check whether an atom E is a member of a list which is not necessarily linear.
c) Write a function to determine the list of all sublists of a given list, on any level.
   A sublist is either the list itself, or any element that is a list, at any level. Example:
   (1 2 (3 (4 5) (6 7)) 8 (9 10)) => 5 sublists :
   ((1 2 (3 (4 5) (6 7)) 8 (9 10)) (3 (4 5) (6 7)) (4 5) (6 7) (9 10))
d) Write a function to transform a linear list into a set.

2.
a) Write a function to return the product of two vectors.
https://en.wikipedia.org/wiki/Dot_product
b) Write a function to return the depth of a list. Example: the depth of a linear list is 1.
c) Write a function to sort a linear list without keeping the double values.
d) Write a function to return the intersection of two sets.

3.
a) Write a function that inserts in a linear list a given atom A after the 2nd, 4th, 6th, ... element.
b) Write a function to get from a given list the list of all atoms, on any
   level, but reverse order. Example:
   (((A B) C) (D E)) ==> (E D C B A)
c) Write a function that returns the greatest common divisor of all numbers in a nonlinear list.
d) Write a function that determines the number of occurrences of a given atom in a nonlinear list.

4.
a) Write a function to return the sum of two vectors.
b) Write a function to get from a given list the list of all atoms, on any
   level, but on the same order. Example:
   (((A B) C) (D E)) ==> (A B C D E)
c) Write a function that, with a list given as parameter, inverts only continuous
   sequences of atoms. Example:
   (a b c (d (e f) g h i)) ==> (c b a (d (f e) i h g))
d) Write a list to return the maximum value of the numerical atoms from a list, at superficial level.

5.
a) Write twice the n-th element of a linear list. Example: for (10 20 30 40 50) and n=3 will produce (10 20 30 30 40 50).
b) Write a function to return an association list with the two lists given as parameters.
   Example: (A B C) (X Y Z) --> ((A.X) (B.Y) (C.Z)).
c) Write a function to determine the number of all sublists of a given list, on any level.
   A sublist is either the list itself, or any element that is a list, at any level. Example:
   (1 2 (3 (4 5) (6 7)) 8 (9 10)) => 5 lists:
   (list itself, (3 ...), (4 5), (6 7), (9 10)).
d) Write a function to return the number of all numerical atoms in a list at superficial level.

6.
a) Write a function to test whether a list is linear.
b) Write a function to replace the first occurence of an element E in a given list with an other element O.
c) Write a function to replace each sublist of a list with its last element.
A sublist is an element from the first level, which is a list.
Example: (a (b c) (d (e (f)))) ==> (a c (e (f))) ==> (a c (f)) ==> (a c f)
(a (b c) (d ((e) f))) ==> (a c ((e) f)) ==> (a c f)
d) Write a function to merge two sorted lists without keeping double values.

7.
a) Write a function to eliminate the n-th element of a linear list.
b) Write a function to determine the successor of a number represented digit by digit as a list, without transforming the representation of the number from list to number. Example: (1 9 3 5 9 9) --> (1 9 3 6 0 0)
c) Write a function to return the set of all the atoms of a list.
Exemplu: (1 (2 (1 3 (2 4) 3) 1) (1 4)) ==> (1 2 3 4)
d) Write a function to test whether a linear list is a set.

8.
a) Write a function to return the difference of two sets.
b) Write a function to reverse a list with its all sublists, on all levels.
c) Write a function to return the list of the first elements of all list elements of a given list with an odd number of elements at superficial level. Example:
(1 2 (3 (4 5) (6 7)) 8 (9 10 11)) => (1 3 9).
d) Write a function to return the sum of all numerical atoms in a list at superficial level.

9.
a) Write a function that merges two sorted linear lists and keeps double values.
b) Write a function to replace an element E by all elements of a list L1 at all levels of a given list L.
c) Write a function to determines the sum of two numbers in list representation, and returns the corresponding decimal number, without transforming the representation of the number from list to number.
d) Write a function to return the greatest common divisor of all numbers in a linear list.

10.
a) Write a function to return the product of all the numerical atoms from a list, at superficial level.
b) Write a function to replace the first occurence of an element E in a given list with an other element O.
c) Write a function to compute the result of an arithmetic expression memorised
in preorder on a stack. Examples:
(+ 1 3) ==> 4 (1 + 3)
(+ * 2 4 3) ==> 11 [((2 * 4) + 3)
(+ * 2 4 - 5 * 2 2) ==> 9 ((2 * 4) + (5 - (2 * 2))
d) Write a function to produce the list of pairs (atom n), where atom appears for n times in the parameter list. Example:
(A B A B A C A) --> ((A 4) (B 2) (C 1)).

11.
a) Determine the least common multiple of the numerical values of a nonlinear list.
b) Write a function to test if a linear list of numbers has a "mountain" aspect (a list has a "mountain" aspect if the items increase to a certain point and then decreases.
Eg. (10 18 29 17 11 10). The list must have at least 3 atoms to fullfil this criteria.
c) Remove all occurrences of a maximum numerical element from a nonlinear list.
d) Write a function which returns the product of numerical even atoms from a list, to any level.

12.
a) Write a function to return the dot product of two vectors. https://en.wikipedia.org/wiki/Dot_product
b) Write a function to return the maximum value of all the numerical atoms of a list, at any level.
c) All permutations to be replaced by: Write a function to compute the result of an arithmetic expression memorised
  in preorder on a stack. Examples:
  (+ 1 3) ==> 4 (1 + 3)
  (+ * 2 4 3) ==> 11 [((2 * 4) + 3)
  (+ * 2 4 - 5 * 2 2) ==> 9 ((2 * 4) + (5 - (2 * 2))
d) Write a function to return T if a list has an even number of elements on the first level, and NIL on the contrary case, without counting the elements of the list.

13.
a) A linear list is given. Eliminate from the list all elements from N to N steps, N-given.
b) Write a function to test if a linear list of integer numbers has a "valley" aspect (a list has a valley aspect if the items decrease to a certain point and then increase. Eg. 10 8 6 17 19 20). A list must have at least 3 elements to fullfill this condition.
c) Build a function that returns the minimum numeric atom from a list, at any level.
d) Write a function that deletes from a linear list of all occurrences of the maximum element.

14.
a) Write a function to return the union of two sets.
b) Write a function to return the product of all numerical atoms in a list, at any level.
c) Write a function to sort a linear list with keeping the double values.
d) Build a list which contains positions of a minimum numeric element from a given linear list.

15.
a) Write a function to insert an element E on the n-th position of a linear list.
b) Write a function to return the sum of all numerical atoms of a list, at any level.
c) Write a function to return the set of all sublists of a given linear list. Ex. For list ((1 2 3) ((4 5) 6)) => ((1 2 3) (4 5) ((4 5) 6))
d) Write a function to test the equality of two sets, without using the difference of two sets.