Lisp Programming (15 marks)

This question is about lisp programming. You can use any builtin/pre-defined functions that are available in SBCL.

(i) [7 marks] Write a Lisp function `(defun exam0 (L) ...)`, where `L` is a (nonempty) list of sublists of numbers, and the function returns the sublist of `L` whose sum is the largest among all sublists in `L`. E.g., If `L = ((2 3) (1) (8 9) (4 6 5))`, the function should return the sublist (8 9). If multiple sublists have the same largest sum, any of them may be returned. We assume that a sublist is nonempty.

(ii) [8 marks] Define a Lisp function `(defun exam1 (L) ...)`, where `L` is a list of sublists of atoms, and the function returns a sublist of `L` which is a superlist of any other sublist in `L`. A list L is a superlist of another list L' if every atom in L' is also an atom in L. E.g., `(a b c)` is a superlist of `(b a)`, `{}`, or `(a c b)`, to name a few. If such a superlist does not exist, the function returns NIL.