Functional and logic programming
- ## written exam -

Important:
1. Subjects are graded as follows: By default - 1p; A – 2p; B - 4p; C - 3p.
2. Prolog problems will be resolved using SWI Prolog. The following are required: (1) explanation of the code and of the reasoning behind it; (2) recursive model that solves the problem, for all the predicates used; (3) specification of every predicate (parameters and their meaning, flow model, type of the predicate - deterministic/non-deterministic).
3. Lisp problems will be resolved using Common Lisp. The following are required: (1) explanation of the code and of the reasoning behind it; (2) recursive model that solves the problem, for each function used; (3) specification of every function (parameters and their meaning).

A. The following function definition in LISP is given
```
(DEFUN F(L)
       (COND
              ((NULL L) 0)
              (> (F (CDR L)) 2) (+ (F (CDR L)) (CAR L)))
              (T (+ (F (CDR L)) 1))
       )
)
```

Rewrite the definition in order to avoid the repeated recursive call **(F (CDR L))**. Do NOT redefine the function. Do NOT use SET, SETQ, SETF. Justify your answer.

B. Write a PROLOG program that generates the list of permutations of the set 1..N, having the property that the absolute value of the difference between 2 consecutive values from the permutation is >=2. Write the mathematical models and flow models for the predicates used. For example, for N=4 ⇒ [[3,1,4,2], [2,4,1,3]] (not necessarily in this order).

C. An n-ary tree is represented in Lisp as (node subtree1 subtree2 ...). Write a Lisp program to return the *height* of a node of a tree. **A MAP function shall be used.**
Example for the tree (a (b (g)) (c (d (e)) (f)))
a) nod=e => the height is 0   **b)** nod=v => the height is -1   **c)** nod=c => the height is 2.