Functional and logic programming
- **written exam -**

Important:

1. Subjects are graded as follows: of - 1p; A – 1.5p; B - 2.5p; C - 2.5p; D - 2.5p.
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 (CAR L)) 2) (+ (CAR L) (F (CDR L))))
                (T (F (CAR L)))
        )
)
```

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

B. Given a heterogeneous list composed of numbers and numeric linear lists, write a SWI-PROLOG program that calculates the difference between the greatest number from the sublists and the smallest number from the superficial level of the list. We assume that the input list contains at least one sublist and at least one number at the superficial level, but we do not know the minimum/maximum possible value for the numbers in the list/sublists. **For example**, for the list [[4, 2, 8], 7, 2, -3, [6, 9, 11, 2], 4], the result will be 14 [11 - [-3]].

C. Write a PROLOG program that generates the list of all subsets with values between the [a, b] interval such that the sum of elements from each subset is an odd value. Write the mathematical models and flow models for the predicates used. For example, for **a**=2 and **b**=4 ⇒ [[2,3],[3,4],[2,3,4]] (not necessarily in this order).

D. Given a nonlinear list, write a Lisp function to replace all the odd values from even levels with their natural successor. The superficial level is assumed 1. **A MAP function shall be used.** ***Example*** for the list (1 s 4 (3 f (7))) the result is (1 s 4 (4 f (7))).