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)
                ((> (CAR L) 0)
                        (COND
                                ((> (CAR L) (F (CDR L))) (CAR L))
                                (T (F (CDR L)))
                        )
                (T (F (CDR L)))
        )
)
```

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. Given a binary tree in which the nodes contain numerical information and given that the binary tree is represented as a list in which each node is followed by a number (0,1 or 2) that represents the number of children of that node, write a SWI-Prolog program that computes the sum of the first element on each level. For example, for the list [13, 2, 9, 2, 5, 0, 3, 2, 11, 0, 6, 1, 3, 0, 2, 1, 7, 1, 9, 1, 8, 2, 4, 0, 2, 1, 10, 0] the result will be 55.

C. Write a PROLOG program that generates the list or all arrangements of **k** elements from a list of integer numbers, for which the product of the elements is less than a value **V** given. Write the mathematical models and flow models for the predicates used. For example, for the list [1, 2, 3], **k**=2 and **V**=7 ⇒ [[1,2],[2,1],[1,3],[3,1],[2,3],[3,2]] (not necessarily in this order).

D. Given a nonlinear list, write a Lisp function to return the list with all occurrences of the element **e** replaced by the value **e1**. **A MAP function shall be used.**

Example **a)** if the list is (1 (2 A (3 A)) (A)), **e** is A and **e1** is B => (1 (2 B (3 B)) (B))

b) if the list is (1 (2 (3))) and **e** is A => (1 (2 (3)))