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
            ((ATOM L) -1)
            ((> (F (CAR L)) 0) (+ (CAR L) (F (CAR L)) (F (CDR L))))
            (T (F (CDR 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. Write a PROLOG program that determines from a list made of integer numbers, the list of subsets with at least 2 elements, composed of numbers in strictly increasing order. Write the mathematical models and flow models for the predicates used. For example for the list [1, 8, 6, 4] ⇒ [[1,8],[1,6],[1,4],[6,8],[4,8],[4,6],[1,4,6],[1,4,8],[1,6,8],[4,6,8],[1,4,6,8]] (not necessarily in this order).

C. 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)))