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) NIL)
                (> (F (CAR L)) 0) (CONS (F (CAR L)) (F (CDR L))))
                (T (F (CAR L)))
        )
)
```

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

B. Given a numerical linear list, write a SWI-Prolog program that builds a list of lists such that: the first element is the initial list, and then, each element is represented by the previous list in which the increasing sequences of numbers were reversed. The last element of the list will be the list in which all elements are sorted increasingly. For example, for the list [1,3,6,5,2] the result will be: [[1, 3, 6, 5, 2], [6, 3, 1, 5, 2], [6, 3, 5, 1, 2], [6, 5, 3, 2, 1]].

C. For a given value N, generate the list of all permutations with elements N, N+1, ..., 2*N-1 with the property that the absolute value between two consecutive values from the permutation is <=2. Write the mathematical models and flow models for the predicates used.

D. Given a nonlinear list, write a Lisp function to return the list with all atoms on the level **k** replaced by 0. The superficial level is assumed 1. **A MAP function shall be used.**
Example for the list (a (1 (2 b)) (c (d))) **a)** k=2 => (a (0 (2 b)) (0 (d)))
b) k=1 => (0 (1 (2 b)) (c (d))) **c)** k=4 => the list does not change