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)
                ((LISTP (CAR L)) (APPEND (F (CAR L)) (F (CDR L)) (CAR (F (CAR L)))))
                (T (LIST(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 numerical linear list, write a SWI-PROLOG program that applies a stable sorting on this list and sorts the elements in increasing order by the reminder of the division with 3. For example, for the list [10, 5, 6, 12, 7, 3, 20, 30] the result will be [6, 12, 3, 30, 10, 7, 5, 20]. (Obs: stable sort means that elements that are equal elements will remain in the same order as in the initial list, for example 6 and 12).

C. Write a PROLGO program that generates the list of arrangements of k elements from a list of integer numbers, having the given product P. Write the mathematical models and flow models for the predicates used. For example, for the list [2, 5, 3, 4, 10], k=2 and P=20 ⇒ [[2,10],[10,2],[5,4],[4,5]] (not necessarily in this order).

D. Given a nonlinear list, write a Lisp function to return the list with all atoms on 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