

## 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.** Given the following PROLOG predicate definition **f(integer, integer)**, with the flow model (i, o):

f(100, 1):-!.

f(K,X):-K1 is K+1, **f(K1,Y)**, Y>1, !, K2 is K1-1, X is K2+Y.

f(K,X):-K1 is K+1, **f(K1,Y)**, Y>0.5, !, X is Y.

f(K,X):-K1 is K+1, **f(K1,Y)**, X is Y-K1.

Rewrite the definition in order to avoid the recursive call **f(J,V)** in all clauses. Do NOT redefine the predicate. Justify your answer.

**B.** Given a nonlinear list containing both numerical and non-numerical atoms, write a LISP program that computes the greatest common divisor of the numbers situated between two non-numerical atoms (neighbours of a number are not level dependent). **For example**, for the list (A B 12 (5 D (A F (15 B) D (5 F) 4)) C 9) the numbers for which we need to compute the greatest common divisor are: 15 (between F and B), 5 (between D and F) and 4 (between F and C), and the result will be 60. You are not allowed to use the predefined *gcd* Lisp function.

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

**D.** An n-ary tree is represented in Lisp as ( node subtree1 subtree2 ...). Write a function to replace all nodes on odd levels with a given value **e**. The root level is assumed zero. **A MAP function shall be used.**

**Example** for the tree (a (b (g)) (c (d (e)) (f))) and **e=h** => (a (h (g)) (h (d (h)) (h)))