

## 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(1, 1):-!.

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

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-1.

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 nonnumerical atoms, write a LISP program that builds a list containing as sublists non-numerical atoms on each level of the initial list (the first sublist of the result contains non-numerical atoms on the first level, the second sublist the non-numerical atoms from the second level etc.). For example, for the list (A B 12 (5 D (A F (10 B) D (5 F) 1)) C 9 (F 4 (D) 9 (F (H 7) K) (P 4)) X) the result will be ((A B C X) (D F) (A F D D F K P) (B F H)).

**C.** Write a PROLOG program that generates the list of all subsets with value of sum for each subset odd number and also odd numbers of odd values from each subset. Write the mathematical models and flow models for the predicates used. For example, for  $[2,3,4] \Rightarrow [[2,3],[3,4],[2,3,4]]$  not necessarily in this order).

**D.** An n-ary tree is represented in Lisp as ( node subtree1 subtree2 ...). Write a Lisp function to return the list of nodes on the given level **k**. The root level is assumed zero. **A MAP function shall be used.** ***Example*** for the tree (a (b (g)) (c (d (e)) (f)))

**a)** k=2 => (g d)      **b)** k=5 => ()