

## 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.** Let L be a list of numbers and given the following PROLOG predicate definition **f(list, integer)**, with the flow model (i, o):

f([], -1).

f([H|T],S):-H>0, **f(T,S1)**,S1<H,!,S is H.

f([\_|T],S):-**f(T,S1)**, S is S1.

Rewrite the definition in order to avoid the recursive call **f(T,S)** in both 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 calculates the total number of non-numerical atoms at the superficial level of the sublists (including the original list) having the first numerical atom (at any level) an even number. For example, for the list (A B 12 (5 D (A F (10 B) D (5 F) 1)) C 9) the result will be 7.

**C.** Write a PROLOG program that generates the list of all combinations of  $k$  elements with numbers from 1 to  $N$ , with the property that difference between two consecutive numbers from a combination has an even value. Write the mathematical models and flow models for the predicates used. For example, for the  $\mathbf{N}=4$ ,  $\mathbf{k}=2 \Rightarrow [[1,3],[2,4]]$  (not necessarily in this order).

**D.** An n-ary tree is represented in Lisp as ( node subtree1 subtree2 ...). Write a Lisp function to determine the path from the root to a given node. **A MAP function shall be used.**

**Example** for the tree (a (b (g)) (c (d (e)) (f)))

**(a)** nod=e => (a c d e)                      **(b)** nod=v => ()