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 with flow model (i, o):

        f([],-1).
        f([H|T],S):-**f(T,S1)**, S1<1, S is S1-H, !.
        f([_|T],S):-**f(T,S)**.

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 numerical and non-numerical atoms, write a LISP program that replaces non-numerical atoms with the number of occurrences of that atom at the level of the list on which it is located. For example, for the list (F A 12 13 (B 11 (A D 15) C C (F)) 18 11 D (A F) F), the result will be (2 1 12 13 (1 11 (1 1 15) 2 2 (1)) 18 11 1 (1 1) 2).

C. Write a PROLOG program that generates the list of all subsets with N elements, using the elements of a list, such that the sum of elements from a subset is an even number. Write the mathematical models and flow models for the predicates used. For example, for the list L=[1, 3, 4, 2] and N=2 $\Rightarrow$ [[1,3], [2,4]].

D. Given a nonlinear list, write a Lisp function to return the list with all occurrences of the element **e** replaced by the value **e1**. **A MAP function shall be used.**

**Example** **a)** if the list is (1 (2 A (3 A)) (A)), **e** is A and **e1** is B => (1 (2 B (3 B)) (B))
      b) if the list is (1 (2 (3))) and **e** is A => (1 (2 (3)))