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(0, -1):-!.
    f(I,Y):-J is I-1, **f(J,V)**, V>0, !, K is J, Y is K+V.
    f(I,Y):-J is I-1, **f(J,V)**, Y is V+I.

Rewrite the definition in order to avoid the recursive call **f(J,V)** 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 returns the list from which non-numerical atoms are eliminated from 3 to 3 (counting from the left to the right, considering all elements regardless of level). The resulted list will keep the structure of the initial list. 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 the list (A B 12 (5 (A F (10) D (5 F) 1)) 9 (F 4 (D) 9 ((H 7) K) (4)) X).

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] ⇒ [[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 determine the number of nodes on 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 => nr=3 (g d f)   **b)** k=4 => nr=0 ()