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, 0):-!.
        f(I,Y):-J is I-1, **f(J,V)**, V>1, !, K is I-2, Y is K.
        f(I,Y):-J is I-1, **f(J,V)**, Y is V+1.

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 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. Write a PROLOG program that generates the list of all subsets with values between the [a, b] interval such that the sum of elements from each subset is an odd value. Write the mathematical models and flow models for the predicates used. For example, for **a**=2 and **b**=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 replace all nodes on the given level **k** 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) k=2 => (a (b (h)) (c (h (e)) (h)))   **(b)** k=4 => (a (b (g)) (c (d (e)) (f)))