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(100, 1):-!.
f(K,X):-K1 is K+1, f(K1,Y), Y>1, !, K2 is K1-1, X is K2+Y.
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-K1.
```

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 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 arrangements of k elements with the value of sum of all elements from each arrangement equal with a given S, from a list of integers. Write the mathematical models and flow models for the predicates used. For example, for the list [6, 5, 3, 4], **k**=2 and **S**=9⇒ [[6,3],[3,6],[5,4],[4,5]] (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 ()