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([], 0).
f([H|T],S):-f(T,S1),H<S1,!,S is H+S1.
f([_|T],S):-f(T,S1), S is S1+2.
```

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 that contains numerical and non-numerical atoms, write a Lisp program that verifies if the average of the numerical atoms on even levels is equal to the average of the numerical atoms on odd levels. For example, for the list (10 A 10 V (10 B (4 H 5)) (3 (A B (J (1) 5 L)))), the result will be true.

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. Given a nonlinear list, write a Lisp function to return the list with all occurrences of an element **e** removed. **A MAP function shall be used.**
Example **a)** if the list is (1 (2 A (3 A)) (A)) and **e** is A => (1 (2 (3)) NIL)
b) if the list is (1 (2 (3))) and **e** is A => (1 (2 (3)))