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 builds a list with the elements from the initial list, from positions k to k (counting from left to right, considering all elements regardless of level), in reverse order. For example, for the list (A B 12 (5 D (A F (10 B) D (5 F) 1)) C 9) and k = 3 the result is (9 F B A 12).

C. Write a PROLOG program that generates the list of all permutations with the property the absolute value of difference between two consecutive values from each permutation is <=3. Write the mathematical models and flow models for the predicates used. For example, for L=[2,7,5] ⇒ [[2,5,7], [7,5,2]] (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)))