Functional and logic programming
- **written exam -**

Important:

1. Subjects are graded as follows: By default - 1p; A – 2p; B - 4p; C - 3p.
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(50, 1):-!.
f(I,Y):-J is I+1, f(J,S), S<1, !, K is I-2, Y is K.
f(I,Y):-J is I+1, f(J,Y).
```

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. For a given value N, generate the list of all permutations with elements N, N+1, ..., 2*N-1 with the property that the absolute value between two consecutive values from the permutation is <=2. Write the mathematical models and flow models for the predicates used.

C. Given a nonlinear list, write a Lisp function to return the list with all the numerical atoms that are multiple of 3 removed. **A MAP function shall be used.**
Example **a)** if the list is (1 (2 A (3 A)) (6)) => (1 (2 A (A)) NIL)
b) if the list is (1 (2 (C))) => (1 (2 (C)))