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. The following function definition in LISP is given

```
(DEFUN F(N)
        (COND
                ((= N 1) 1)
                (> (F (- N 1)) 2) (- N 2))
                (> (F (- N 1)) 1) (F (- N 1)))
                (T (- (F (- N 1)) 1))
        )
)
```

Rewrite the definition in order to avoid the repeated call **(F (- N 1))**. Do NOT redefine the function. Do NOT use SET, SETQ, SETF. Justify your answer.

B. For two lists that represent numbers digit by digit, we define the operation of rearranging in the following way: we compare the digits and form two new lists, one with the greater digits, and one with the smaller digits. For example, rearranging lists [1,4,2,7] and [8,2,9,6,1] will return [8, 2, 9, 6 ,7] and [1, 4, 2, 1]. Given two lists that contain numbers and sublists of digits, write a SWI-Prolog program that returns two lists with sublists containing the result after rearranging the sublists in the two lists. One of the lists contains the sublists with greater digits and the other the sublists with smaller digits. The two input lists have the same number of sublists, but not necessarily on the same positions. For example, for the lists [1, 2, [6, 2, 4], 6, [9, 9, 1, 1], 17, 9, [5, 3, 8, 1, 9]] and [1, 2, 3, [1, 5], 7, 11, [8, 3], 7, 5, [9, 4, 2, 5], 77] the result will be [[6, 2, 5], [9, 9, 8, 3], [5, 9, 8, 2, 9]] and [[1, 4], [1, 1], [3, 4, 1, 5]].

C. Given a list made of integer numbers, generate using PROLOG the list of arrangements with even number of elements, having the sum an odd number. Write mathematical models and flow models for the predicates used. For example, for the list L=[2,3,4] ⇒ [[2,3],[3,2],[3,4],[4,3]] (not necessarily in this order).

D. An n-ary tree is represented in Lisp as (node subtree1 subtree2 ...). Write a function to replace all nodes on odd levels 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 (h (g)) (h (d (h)) (h)))