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 0) 0)
              (> (F (- N 1)) 1) (- N 2))
              (T (+ (F (- N 1)) 1))
        )
)
```

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

B. Given a linear list composed of numbers, write a SWI-Prolog program that returns the list in which each number that is smaller than its successor in the list is multiplied with two. Repeat this operation until there are no possible changes in the list. For example, for the list [1, 2, 3] the result will be [8, 16, 3].

C. Write a PROLOG program that generates the list of all subsets, each subset having an odd sum of elements and also even number of elements. Write the mathematical models and flow models for the predicates used. For example, for [2,3,4] ⇒ [[2,3,4]].

D. Given a nonlinear list, write a Lisp function to return the list with all atoms on level **k** removed. The superficial level is assumed 1. **A MAP function shall be used.**
Example for the list (a (1 (2 b)) (c (d)))
a) k=2 => (a ((2 b)) ((d))) **b)** k=1 => ((1 (2 b)) (c (d))) **c)** k=4 => the list does not change