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. The following function definition in LISP is given

```
(DEFUN F(L)
        (COND
                ((ATOM L) -1)
                ((> (F (CAR L)) 0) (+ (CAR L) (F (CAR L)) (F (CDR L))))
                (T (F (CDR L)))
            )
)
```

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

B. 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]].

C. An n-ary tree is represented in Lisp as (node subtree1 subtree2 ...). Write a Lisp function to verify whether a node **x** occurs on an even level of the tree. The root level is assumed zero. **A MAP function shall be used.**

Example for the tree (a (b (g)) (c (d (e)) (f))) **a) x**=g => T **b)** x=h => NIL