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(L1 L2)
        (APPEND (F (CAR L1) L2)
                (COND
                    ((NULL L1) (CDR L2))
                    (T (LIST (F (CAR L1) L2) (CAR L2)))
                )
                )
        )
```

Rewrite the definition in order to avoid the double recursive call **(F (CAR L1) L2)**. Do NOT redefine the function. Do NOT use SET, SETQ, SETF. Justify your answer.0

B. Write a PROLOG program that generates the list of all combinations of k elements with the value of sum of each combination even number, from a list of integers. Write the mathematical models and flow models for the predicates used. For example, for the list L[6, 5, 3, 4], **k**=2 ⇒ [[6,4],[5,3]] (not necessarily in this order).

C. 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)))