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 Fct(F L)
        (COND
                ((NULL L) NIL)
                ((FUNCALL F (CAR L)) (CONS (FUNCALL F (CAR L)) (Fct F (CDR L))))
                (T NIL)
        )
)
```

Rewrite the definition in order to avoid the double recursive call **(FUNCALL 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 arrangements of k elements with the value of sum of all elements from each arrangement equal with a given S, from a list of integers. Write the mathematical models and flow models for the predicates used. For example, for the list [6, 5, 3, 4], **k**=2 and **S**=9⇒ [[6,3],[3,6],[5,4],[4,5]] (not necessarily in this order).

C. Write a Lisp function to substitute all numerical values at any level of a given nonlinear list with a given value **e**. **A MAP function shall be used.**
Example, for the list (1 d (2 f (3))), **e**=0 the result is (0 d (0 f (0))).