

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(L)
  (COND
    ((NULL L) 0)
    ((> (F (CAR L)) 1) (F (CDR L)))
    (T (+ (F (CAR L)) (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. Given a heterogeneous list composed of numbers and nonempty numerical linear lists, write a SWI-Prolog program that builds a list with the minimum values from those sublists for which the sum of the elements is a prime number. The resulted list will contain elements in reverse order of the initial input list. For example, for the list `[[4, 1, 18], 7, 2, -3, [6, 9, 11, 3], 4, [5, 9, 19]]`, the result will be `[3, 1]`.

C. Write a PROLOG program that generates the list of all subsets with value of sum for each subset odd number and also odd numbers of odd values from each subset. Write the mathematical models and flow models for the predicates used. For example, for $[2,3,4] \Rightarrow [[2,3],[3,4],[2,3,4]]$ not necessarily in this order).

D. An n-ary tree is represented in Lisp as (node subtree1 subtree2 ...). Write a Lisp function to determine the path from the root to a given node. **A MAP function shall be used.**

Example for the tree (a (b (g)) (c (d (e)) (f)))

(a) nod=e => (a c d e) **(b)** nod=v => ()