

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) NIL)
    ((LISTP (CAR L)) (APPEND (F (CAR L)) (F (CDR L)) (CAR (F (CAR L)))))
    (T (LIST(CAR 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 two lists composed of integer numbers and sublists of integer numbers, write a SWI-Prolog program that returns a list that contains, for each possible pair of sublists (one sublist from the first list and one from the second), the product of the maximum elements. For example, for the following two lists of sublists [1,2, [4,2], 6, [3,2]] and [1,2,3,[5,6],8, 5,[12,3], 4,1,[3,8]] the result will be (not necessarily in this order): [24, 48, 32, 18, 36, 24].

C. 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 $S=9 \Rightarrow [[6,3],[3,6],[5,4],[4,5]]$ (not necessarily in this order).

D. Write a Lisp function to substitute an element **e** by other element **e1** at all odd levels of a nonlinear list. The superficial level is assumed 1. **A MAP function shall be used.**

Example, for the list (1 d (2 d (d))), **e**=d and **e1**=f the result is (1 f (2 d (f))).