

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. Let **G** be LISP function and given the following definition

```
(DEFUN F(L)
  (COND
    ((NULL L) 0)
    (> (G L) 2) (+(G L) (F (CDR L))))
    (T (G L))
  )
)
```

Rewrite the definition in order to avoid the repeated call **(G 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 with values between the $[a, b]$ interval such that the sum of elements from each subset is an odd value. Write the mathematical models and flow models for the predicates used. For example, for $a=2$ and $b=4 \Rightarrow [[2,3],[3,4],[2,3,4]]$ (not necessarily in this order).

C. Given a nonlinear list, write a Lisp function to return the list with all atoms on level **k** removed. The superficial level is assumed 1. **A MAP function shall be used.**

Example for the list (a (1 (2 b)) (c (d)))

a) k=2 => (a ((2 b)) ((d))) **b)** k=1 => ((1 (2 b)) (c (d))) **c)** k=4 => the list does not change

C. Given a nonlinear list, write a Lisp function to return the list with all occurrences of an element **e** removed. **A MAP function shall be used.**

Example **a)** if the list is (1 (2 A (3 A)) (A)) and **e** is A => (1 (2 (3)) NIL)

b) if the list is (1 (2 (3))) and **e** is A => (1 (2 (3)))