

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. Given the following PROLOG predicate definition **f(list, integer)**, with the flow model (i, o):

`f([], -1):-!`

`f([_|T], Rez):- f(T,S), S<1, !, Y is S+2.`

`f([H|T], Rez):- f(T,S), S<0, !, Y is S+H.`

`f([_|T], Rez):- f(T,S), Y is S.`

Rewrite the definition in order to avoid the recursive call **f(T,S)** in all clauses. Do NOT redefine the predicate. Justify your answer.

B. Given a nonlinear list that contains numerical and non-numerical atoms, write a Lisp program that concatenates each linear (sub)list in which the number of the numerical atoms is equal to the number of nonnumerical atoms, with the list it belongs to. Repeat this process until the list cannot be modified anymore. For example, for the list (A B (4 A 3) 11 (5 (A (B 3) (C 10) 1) (4 A) X Y Z)5) the result will be (A B (4 A 3) 11 (5 A B 3 C 10 1 4 A X Y Z) 5).

C. Write a PROLOG program that generates the list of all arrangements of **k** elements from a list of integer numbers, for which the product of the elements is less than a value **V** given. Write the mathematical models and flow models for the predicates used. For example, for the list [1, 2, 3], **k**=2 and **V**=7 \Rightarrow [[1,2],[2,1],[1,3],[3,1],[2,3],[3,2]] (not necessarily in this order).

D. Given a nonlinear list, write a Lisp function to replace all the odd values from even levels with their natural successor. The superficial level is assumed 1. **A MAP function shall be used.**

Example for the list (1 s 4 (3 f (7))) the result is (1 s 4 (4 f (7))).