

Programare logică și funcțională

- examen scris -

Notă

1. Subiectele se notează astfel: of - 1p; A - 1.5p; B - 2.5p; C - 2.5p; D - 2.5p.
2. Problemele Prolog vor fi rezolvate în SWI Prolog. Se cere: (1) explicarea codului și a raționamentului; (2) modelul recursiv de rezolvare, pentru fiecare predicat folosit; (3) specificarea fiecărui predicat (semnificația parametrilor, model de flux, tipul predicatului - determinist/nedeterminist).
3. Problemele Lisp vor fi rezolvate în Common Lisp. Se cere: (1) explicarea codului și a raționamentului; (2) modelul recursiv de rezolvare, pentru fiecare funcție folosită; (3) specificarea fiecărei funcții (semnificația parametrilor).

A. Fie următoarea definiție de predicat PROLOG **f(integer, integer)**, având modelul de flux (i, o):

f(50, 1):-!.
f(I,Y):-J is I+1, **f(J,S)**, S<1, !, K is I-2, Y is K.

f(I,Y):-J is I+1, **f(J,Y)**.

Rescrieți această definiție pentru a evita apelul recursiv **f(J,V)** în ambele clauze. Nu redefiniți predicatul. Justificați răspunsul.

- B.** Dându-se o listă neliniară care conține atomi numerici și nenumeriți, se cere un program Lisp care concatenează fiecare (sub)listă liniară în care numărul atomilor numerici este egal cu numărul atomilor nenumeriți cu lista în care se află. Se repetă acest proces până lista nu mai poate fi modificată. De exemplu, pentru lista (A B (4 A 3) 11 (5 (A (B 3) (C 10) 1) (4 A) X Y Z)5) rezultatul va fi (A B (4 A 3) 11 (5 A B 3 C 10 1 4 A X Y Z) 5).

- C. Să se scrie un program PROLOG care generează lista submulțimilor formate cu elemente unei liste listă de numere întregi, având suma elementelor număr impar și număr impar de elemente impare. Se vor scrie modelele matematice și modelele de flux pentru predicatele folosite.

Exemplu- pentru lista $[2,3,4] \Rightarrow [[2,3],[3,4],[2,3,4]]$ (nu neapărat în această ordine)

- D. Se consideră o listă neliniară. Să se scrie o funcție LISP care să aibă ca rezultat lista inițială în care toate aparițiile unui element **e** au fost înlocuite cu o valoare **e1**. **Se va folosi o funcție MAP.**

Exemplu

- a)** dacă lista este (1 (2 A (3 A)) (A)) **e** este A și **e1** este B => (1 (2 B (3 B)) (B))
b) dacă lista este (1 (2 (3))) și **e** este A => (1 (2 (3)))