

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):

$f(0, 0):-!$.

$f(I,Y):-J \text{ is } I-1, \text{ f(J,V), } V>1, !, K \text{ is } I-2, Y \text{ is } K.$

$f(I,Y):-J \text{ is } I-1, \text{ f(J,V), } Y \text{ is } V+1.$

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 conținând atât atomi numerici, cât și nenumeriți, se cere un program LISP care să verifice dacă atomii numerici din listă formează o secvență de valori în ordine crescătoare. **De exemplu**, pentru lista `(A B 1 (2 C D) 3 4 (F T 6 10 (A E D) (34) F) 111))` rezultatul este **adevărat** (T), iar pentru lista `(A B 1 (2 C D) 3 4 (F T 6 1 (A E D) (34) F) 111))` rezultatul este **fals** (NIL).

- 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 din care au fost eliminați toți atomii numerici pari situați pe un nivel impar. Nivelul superficial se consideră a fi 1. **Se va folosi o funcție MAP.**

Exemplu

a) dacă lista este (1 (2 A (4 A)) (6)) => (1 (2 A (A)) (6))

b) dacă lista este (1 (2 (C))) => (1 (2 (C)))