

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(list, integer)**, având modelul de flux (i, o):

$f([], -1):-!$.

$f([_|T], Rez):- \underline{f(T,S)}, S<1, !, Y \text{ is } S+2.$

$f([H|T], Rez):- \underline{f(T,S)}, S<0, !, Y \text{ is } S+H.$

$f([_|T], Rez):- \underline{f(T,S)}, Y \text{ is } S.$

Rescrieți această definiție pentru a evita apelul recursiv **f(T,S)** în 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 calculeze cel mai mare divizor comun al numerelor impare de la nivelurile pare ale listei. Nivelul superficial al listei se consideră 1. **De exemplu,** pentru lista (A B 12 (9 D (A F (75 B) D (45 F) 1) 15) C 9), rezultatul va fi 3. Se presupune că există cel puțin un număr impar la un nivel par al listei. Nu se va folosi funcția predefinită *gcd* din Lisp.

- C. Să se scrie un program PROLOG care generează lista submulțimilor de sumă pară, cu elementele unei liste. Se vor scrie modelele matematice și modelele de flux pentru predicatele folosite.

Exemplu pentru lista $L=[2, 3, 4] \Rightarrow [[],[2],[4],[2,4]]$ (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 multipli de 3. **Se va folosi o funcție MAP.**

Exemplu

- a)** dacă lista este (1 (2 A (3 A)) (6)) => (1 (2 A (A)) NIL)
b) dacă lista este (1 (2 (C))) => (1 (2 (C)))