

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):

$f(100, 1):-!$.

$f(K,X):-K1 \text{ is } K+1, \underline{f(K1,Y)}, Y>1, !, K2 \text{ is } K1-1, X \text{ is } K2+Y.$

$f(K,X):-K1 \text{ is } K+1, \underline{f(K1,Y)}, Y>0.5, !, X \text{ is } Y.$

$f(K,X):-K1 \text{ is } K+1, \underline{f(K1,Y)}, X \text{ is } Y-K1.$

Rescrieți această definiție pentru a evita apelul recursiv **f(J,V)** 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 returneze lista din care să se elimine atomii nenumeriți din 3 în 3 (numărarea se va face de la stânga spre dreapta, considerând toate elementele). Lista rezultată va păstra structura listei inițiale. **De exemplu**, pentru lista (A B 12 (5 D (A F (10 B) D (5 F) 1)) C 9 (F 4 (D) 9 (F (H 7) K) (P 4)) X) rezultatul va fi lista (A B 12 (5 (A F (10) D (5 F) 1)) 9 (F 4 (D) 9 ((H 7) K) (4)) X).

- C. Să se scrie un program PROLOG care generează lista submulțimilor cu suma număr impar, cu valori din intervalul $[a, b]$. Se vor scrie modelele matematice și modelele de flux pentru predicatele folosite.

Exemplu pentru $a=2$ și $b=4 \Rightarrow [[2,3],[3,4],[2,3,4]]$ (nu neapărat în această ordine)

- D. Un arbore n-ar se reprezintă în LISP astfel (nod subarbore1 subarbore2). Se cere să se verifice dacă un nod **x** apare pe un nivel par în arbore. Nivelul rădăcinii se consideră a fi 0. **Se va folosi o funcție MAP.**

Exemplu pentru arborele (a (b (g)) (c (d (e)) (f)))

a) $x=g \Rightarrow T$ **b)** $x=h \Rightarrow NIL$