

Programare logică și funcțională

- examen scris -

Notă

1. Subiectele se notează astfel: of - 1p; A - 2p; B - 4p; C - 3p.
2. Problema Prolog (B) vor fi rezolvată 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. Problema Lisp (C) va fi rezolvată 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 is K+1, **f(K1,Y)**, Y>1, !, K2 is K1-1, X is K2+Y.

f(K,X):-K1 is K+1, **f(K1,Y)**, Y>0.5, !, X is Y.

f(K,X):-K1 is K+1, **f(K1,Y)**, X is Y-K1.

Rescrieți această definiție pentru a evita apelul recursiv **f(J,V)** în clauze, fără a redefini logica clauzelor. Justificați răspunsul.

B. 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)

C. Un arbore n-ar se reprezintă în LISP astfel (nod subarbore1 subarbore2). Se cere să se determine înălțimea unui nod în arbore. **Se va folosi o funcție MAP.**

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

a) nod=e => înălțimea e 0 **b)** nod=v => înălțimea e -1 **c)** nod=c => înălțimea e 2