

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 L o listă numerică și următoarea definiție de predicat PROLOG având modelul de flux (i, o):

$f([], -1)$.

$f([H|T], S) :- f(T, S1), S1 < 1, S \text{ is } S1 - H, !$.

$f([_|T], S) :- f(T, S)$.

Rescrieți această definiție pentru a evita apelul recursiv $f(T, S)$ în ambele clauze, fără a redefini logica clauzelor. Justificați răspunsul.

B. Dându-se o listă formată din numere întregi, să se genereze lista submulțimilor cu **k** elemente în progresie aritmetică. Se vor scrie modelele matematice și modelele de flux pentru predicatele folosite.

Exemplu- pentru lista $L=[1,5,2,9,3]$ și $k=3 \Rightarrow [[1,2,3],[1,5,9],[1,3,5]]$ (nu neapărat în această ordine)

C. Un arbore n-ar se reprezintă în LISP astfel (nod subarbore1 subarbore2). Se cere să se determine calea de la rădăcină către un nod dat. **Se va folosi o funcție MAP.**

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

a) nod=e => (a c d e) **b)** nod=v => ()