

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 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. Nu redefiniți predicatul. Justificați răspunsul.

- B.** Dându-se o listă neliniară conținând atât atomi numerici cât și nenumeriți, se cere un program LISP care construiește o listă cu elementele listei inițiale, din k în k (numărarea se face de la stânga la dreapta, considerând toate elementele), în ordine inversă. **De exemplu**, pentru lista (A B 12 (5 D (A F (10 B) D (5 F) 1)) C 9) și $k = 3$ rezultatul este (9 F B A 12). Nu este permisă utilizarea funcției predefinite *reverse* din Lisp.

- C. Să se scrie un program PROLOG care generează lista aranjamentelor de **k** elemente dintr-o listă de numere întregi, având o sumă **S** dată. Se vor scrie modelele matematice și modelele de flux pentru predicatele folosite.

Exemplu- pentru lista [6, 5, 3, 4], **k**=2 și **S**=9 \Rightarrow [[6,3],[3,6],[5,4],[4,5]] (nu neapărat în această ordine)

- D. Un arbore n-ar se reprezintă în LISP astfel (nod subarbore1 subarbore2). Se cere să se determine lista nodurilor de pe nivelurile pare din arbore (în ordinea nivelurilor 0, 2, ...). Nivelul rădăcinii se consideră 0. **Se va folosi o funcție MAP.**
- Exemplu** pentru arborele (a (b (g)) (c (d (e)) (f))) => (a g d f)