

# 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):

$f([], 0).$

$f([H|T], S) :- f(T, S1), S1 \geq 2, !, S \text{ is } S1 + H.$

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

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 care conține atomi numerici și nenumeriți, se cere un program Lisp care verifică dacă media atomilor numerici de pe niveluri pare este egală cu media atomilor numerici de pe niveluri impare. De exemplu, pentru lista (10 A 10 V (10 B (4 H 5)) ( 3 (A B (J (1) 5 L)))) rezultatul va fi true.

- C. Dându-se o listă formată din numere întregi, să se genereze în PROLOG lista submulțimilor cu număr par de elemente. Se vor scrie modelele matematice și modelele de flux pentru predicatele folosite.

**Exemplu** pentru lista  $L=[2,3,4] \Rightarrow [[],[2,3],[2,4],[3,4]]$  (nu neapărat în această ordine)

D. 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