

# 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 **f(list, integer)**, având modelul de flux (i, o):

$f([], 0)$ .

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

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

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 în PROLOG lista permutărilor având proprietatea că valoarea absolută a diferenței dintre două valori consecutive din permutare este  $\leq 3$ . Se vor scrie modelele matematice și modelele de flux pentru predicatele folosite.

**Exemplu**- pentru lista  $L=[2,7,5] \Rightarrow [[2,5,7], [7,5,2]]$  (nu neapărat în această ordine)

**C.** Se consideră o listă neliniară. Să se scrie o funcție LISP care să aibă ca rezultat lista inițială în care toate aparițiile unui element **e** au fost înlocuite cu o valoare **e1**. **Se va folosi o funcție MAP.**

**Exemplu**

- a)** dacă lista este (1 (2 A (3 A)) (A)) **e** este A și **e1** este B => (1 (2 B (3 B)) (B))  
**b)** dacă lista este (1 (2 (3))) și **e** este A => (1 (2 (3)))