

# 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):

$f([], -1).$

$f([H|T], S) :- f(T, S1), S1 > 0, !, S \text{ is } S1 + H.$

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

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 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)

**C.** Un arbore n-ar se reprezintă în LISP astfel (nod subarbore1 subarbore2 .....). Se cere să se determine numărul de noduri de pe nivelul **k**. Nivelul rădăcinii se consideră 0. **Se va folosi o funcție MAP.**

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

**a)**  $k=2 \Rightarrow nr=3$  (g d f)    **b)**  $k=4 \Rightarrow nr=0$  ()