

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

f([], 0).

f([H|T], S):-**f(T, S1)**, S1<H,!,S is H.

f([\_|T], S):-**f(T, S1)**, S is S1.

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 să înlocuiască fiecare atom nenumeric cu numărul de apariții ale atomului la nivelul pe care se află. **De exemplu**, pentru lista (F A 12 13 (B 11 (A D 15) C C (F)) 18 11 D (A F) F), rezultatul va fi (2 1 12 13 (1 11 (1 1 15) 2 2 (1)) 18 11 1 (1 1) 2).

- C. Să se scrie un program PROLOG care generează lista combinațiilor de **k** elemente cu numere de la 1 la **N**, având diferența între două numere consecutive din combinație număr par. Se vor scrie modelele matematice și modelele de flux pentru predicatele folosite.

**Exemplu**- pentru **N=4**, **k=2**  $\Rightarrow$   $[[1,3],[2,4]]$  (nu neapărat în această ordine)

- D. Se consideră o listă neliniară. Să se scrie o funcție LISP care să aibă ca rezultat lista inițială din care au fost eliminați toți atomii nenumeriți de pe nivelurile pare (nivelul superficial se consideră 1). **Se va folosi o funcție MAP.**

**Exemplu** pentru lista (a (1 (2 b)) (c (d))) rezultă (a (1 (2 b)) ((d)))