

# 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).
4. 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):

$f([], -1)$ .

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

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

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 returneze lista din care să se elimine atomii nenumeriți din 3 în 3 (numărarea se va face de la stânga spre dreapta, considerând toate elementele). Lista rezultată va păstra structura listei inițiale. **De exemplu**, pentru lista (A B 12 (5 D (A F (10 B) D (5 F) 1)) C 9 (F 4 (D) 9 (F (H 7) K) (P 4)) X) rezultatul va fi lista (A B 12 (5 (A F (10) D (5 F) 1)) 9 (F 4 (D) 9 ((H 7) K) (4)) X).

- C. Să se scrie un program PROLOG care generează lista permutărilor mulțimii  $1..N$ , cu proprietatea că valoarea absolută a diferenței între 2 valori consecutive din permutare este  $\geq 2$ . Se vor scrie modelele matematice și modelele de flux pentru predicatele folosite.

**Exemplu**- pentru  $N=4 \Rightarrow [[3,1,4,2], [2,4,1,3]]$  (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 numerici pari situați pe un nivel impar. Nivelul superficial se consideră a fi 1. **Se va folosi o funcție MAP.**

**Exemplu**

**a)** dacă lista este (1 (2 A (4 A)) (6)) => (1 (2 A (A)) (6))

**b)** dacă lista este (1 (2 (C))) => (1 (2 (C)))