

# 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 următoarea definiție de predicat PROLOG  **$f(\text{integer}, \text{integer})$** , având modelul de flux (i, o):

$f(1, 1):-!$ .

$f(K,X):-K1 \text{ is } K-1, \text{  $f(K1,Y)$ , } Y>1, !, K2 \text{ is } K1-1, X \text{ is } K2.$

$f(K,X):-K1 \text{ is } K-1, \text{  $f(K1,Y)$ , } Y>0.5, !, X \text{ is } Y.$

$f(K,X):-K1 \text{ is } K-1, \text{  $f(K1,Y)$ , } X \text{ is } Y-1.$

Rescrieți această definiție pentru a evita apelul recursiv  **$f(J,V)$**  în clauze. Nu redefiniți predicatul. Justificați răspunsul.

- B.** Dându-se o listă neliniară care conține atomi numerici și nenumeriți, se cere un program Lisp care construiește o listă care conține doar atomii numerici, alternativ un număr par urmat de unul impar. Numerele impare sunt în câte o sublistă. Numerele pare și impare sunt în aceeași ordine relativă ca în lista inițială. Se garantează că lista inițială conține un număr egal de numere impare și pare. De exemplu, pentru lista (A B (4 A 2 ) 11 (5 (A (B 20) C 10) (1(2(3(4)5)6)7 7) X Y Z)) rezultatul va fi (4 (11) 2 (5) 20 (1) 10 (3) 2 (5) 4 (7) 6 (7)).

- C. Pentru o valoare **N** dată, să se genereze lista permutărilor cu elementele  $N, N+1, \dots, 2*N-1$  având proprietatea că valoarea absolută a diferenței dintre două valori consecutive din permutare este  $\leq 2$ . Se vor scrie modelele matematice și modelele de flux pentru predicatele folosite.

- 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 multipli de 3. **Se va folosi o funcție MAP.**

**Exemplu**

- a)** dacă lista este (1 (2 A (3 A)) (6)) => (1 (2 A (A)) NIL)  
**b)** dacă lista este (1 (2 (C))) => (1 (2 (C)))