

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

$f(0, 0):-!$ .

$f(I,Y):-J \text{ is } I-1, \text{ f(J,V), } V>1, !, K \text{ is } I-2, Y \text{ is } K.$

$f(I,Y):-J \text{ is } I-1, \text{ f(J,V), } Y \text{ is } V+1.$

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

- B.** Dându-se o listă neliniară conținând atât atomi numerici cât și nenumeriți, să se scrie un program LISP care calculează cel mai mare divizor comun al acelor numere care se află între 2 atomi nenumeriți (vecinii unui număr se consideră la orice nivel). **De exemplu**, pentru lista (A B 12 (5 D (A F (15 B) D (5 F) 4)) C 9) numerele al căror cel mai mare divizor comun trebuie calculat sunt: 15 (între F și B), 5 (între D și F) și 4 (între F și C), iar rezultatul va fi 60. Nu este permisă utilizarea funcției predefinite *gcd* din Lisp.

- C. Să se scrie un program PROLOG care generează lista submulțimilor cu valori din intervalul  $[a, b]$ , având număr par de elemente pare și număr impar de elemente impare. Se vor scrie modelele matematice și modelele de flux pentru predicatele folosite.

**Exemplu**- pentru  $a=2$  și  $b=4 \Rightarrow [[2,3,4]]$

- D. Se consideră o listă neliniară. Să se scrie o funcție care să aibă ca rezultat lista inițială în care atomii de pe nivelul **k** au fost înlocuiți cu 0 (nivelul superficial se consideră 1). **Se va folosi o funcție MAP.**

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

**a)** k=2 => (a (0 (2 b)) (0 (d)))    **b)** k=1 => (0 (1 (2 b)) (c (d)))    **c)** k=4 => lista nu se modifică