

# 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(50, 1):-!.

f(I,Y):-J is I+1, **f(J,S)**, S<1, !, K is I-2, Y is K.

f(I,Y):-J is I+1, **f(J,Y)**.

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, se cere un program LISP care construiește o listă cu elementele listei inițiale, din  $k$  în  $k$  (numărarea se face de la stânga la dreapta, considerând toate elementele), în ordine inversă. **De exemplu**, pentru lista (A B 12 (5 D (A F (10 B) D (5 F) 1)) C 9) și  $k = 3$  rezultatul este (9 F B A 12). Nu este permisă utilizarea funcției predefinite *reverse* din Lisp.

- C. Dându-se o listă formată din numere întregi, să se genereze în PROLOG lista submulțimilor cu cel puțin **N** elemente având suma divizibilă cu 3. Se vor scrie modelele matematice și modelele de flux pentru predicatele folosite.

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

D. Un arbore n-ar se reprezintă în LISP astfel ( nod subarbore1 subarbore2 .....)

Se cere să se înlocuiască nodurile de pe nivelurile impare din arbore cu o valoare **e** dată. Nivelul rădăcinii se consideră a fi 0. **Se va folosi o funcție MAP.**

**Exemplu** pentru arborele (a (b (g)) (c (d (e)) (f))) și **e=h** => (a (h (g)) (h (d (h)) (h)))