

# 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.** Un arbore n-ar poate fi memorat ca o listă liniară în care fiecare nod este urmat de numărul de descendenți. Dându-se o listă liniară care reprezintă un arbore n-ar, se cere un program Lisp care determină, sub forma unei liste, al k-lea descendent al rădăcinii din arbore. De exemplu, pentru arborele (A 5 B 2 E 0 F 3 G 0 H 0 I 0 C 1 J 1 K 2 L 0 M 0 D 4 N 0 O 0 P 2 R 0 S 1 T 0 Q 0 U 0 V 1 Z 2 T 0 W 0) și  $k = 3$  rezultatul va fi (D 4 N 0 O 0 P 2 R 0 S 1 T 0 Q 0), iar pentru același arbore și  $k = 5$  rezultatul va fi (V 1 Z 2 T 0 W 0).

- C. Să se scrie un program PROLOG care generează lista submulțimilor de sumă pară, cu elementele unei liste. Se vor scrie modelele matematice și modelele de flux pentru predicatele folosite.

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

- D. Un arbore n-ar se reprezintă în LISP astfel (nod subarbore1 subarbore2 .....). Se cere să se verifice dacă un nod **x** apare pe un nivel par în arbore. Nivelul rădăcinii se consideră a fi 0. **Se va folosi o funcție MAP.**

**Exemplu** pentru arborele (a (b (g)) (c (d (e)) (f)))

**a)**  $x=g \Rightarrow T$       **b)**  $x=h \Rightarrow NIL$