

# 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 funcție LISP

```
(DEFUN F(N)
  (COND
    ((= N 1) 1)
    (> (F (- N 1)) 2) (- N 2))
    (> (F (- N 1)) 1) (F (- N 1)))
    (T (- (F (- N 1)) 1))
  )
)
```

Rescrieți această definiție pentru a evita apelul repetat **(F (- N 1))**. Nu redefiniți funcția. Nu folosiți SET, SETQ, SETF. Justificați răspunsul.

- B.** Dându-se o listă eterogenă formată din numere și liste liniare nevide de numere, se cere un program SWI-PROLOG care să calculeze diferența dintre cel mai mic maxim din subliste și cel mai mare dintre valorile minime din subliste. Se presupune că lista de intrare conține cel puțin o sublistă. **De exemplu**, pentru lista `[[4, 2, 18], 7, 2, -3, [6, 9, 11, 3], 4, [5, 9, 19]]` rezultatul va fi 6.

- 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. Se consideră o listă neliniară. Să se scrie o funcție LISP care să aibă ca rezultat lista inițială în care toate aparițiile unui element **e** au fost înlocuite cu o valoare **e1**. **Se va folosi o funcție MAP.**

**Exemplu**

- a)** dacă lista este (1 (2 A (3 A)) (A)) **e** este A și **e1** este B => (1 (2 B (3 B)) (B))  
**b)** dacă lista este (1 (2 (3))) și **e** este A => (1 (2 (3)))