

# 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 0) 0)
    (> (F (- N 1)) 1) (- N 2))
    (T (+ (F (- N 1)) 1))
  )
)
```

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

- B. Dându-se o listă liniară de numere, se cere un program SWI-Prolog care înlocuiește secvențele de numere egale cu suma secvenței. Acest proces trebuie repetat până când nu mai sunt elemente consecutive egale în listă. De exemplu, pentru lista [1, 2, 1, 1, 4, 5, 6, 7, 7, 7, 3, 3, 3, 3, 3, 3, 3, 10], rezultatul va fi [1, 8, 5, 6, 42, 10].

- C. Dându-se o listă formată din numere întregi, să se genereze în PROLOG lista permutărilor având proprietatea că valoarea absolută a diferenței dintre două valori consecutive din permutare este  $\leq 3$ . Se vor scrie modelele matematice și modelele de flux pentru predicatele folosite.

**Exemplu**- pentru lista  $L=[2,7,5] \Rightarrow [[2,5,7], [7,5,2]]$  (nu neapărat în această ordine)

D. Un arbore n-ar se reprezintă în LISP astfel ( nod subarbore1 subarbore2 .....). Se cere să se determine înălțimea unui nod în arbore. **Se va folosi o funcție MAP.**

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

**a)** nod=e => înălțimea e 0      **b)** nod=v => înălțimea e -1      **c)** nod=c => înălțimea e 2