

# 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 formată din numere, se cere un program SWI-PROLOG care să furnizeze lista în care fiecare număr care este mai mic decât succesorul său din listă este înmulțit cu 2. Repetați această operație până când nu mai sunt posibile modificări în listă. **De exemplu**, pentru lista [1, 2, 3] rezultatul va fi [8, 16, 3].

- C. Să se scrie un program PROLOG care generează lista submulțimilor cu **N** elemente, cu elementele unei liste, astfel încât suma elementelor dintr-o submulțime să fie număr par. Se vor scrie modelele matematice și modelele de flux pentru predicatele folosite.

**Exemplu**- pentru lista  $L=[1, 3, 4, 2]$  și  $N=2 \Rightarrow [[1,3], [2,4]]$

D. Se dă o listă neliniară și se cere înlocuirea valorilor numerice pare cu numărul natural succesor. **Se va folosi o funcție MAP.**

**Exemplu** pentru lista (1 s 4 (2 f (7))) va rezulta (1 s 5 (3 f (7))).