

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(L)
  (COND
    ((NULL L) 0)
    ((> (F (CAR L)) 1) (F (CDR L)))
    (T (+ (F (CAR L)) (F (CDR L)))))
  )
)
```

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

- B.** Dându-se 2 liste formate din numere întregi și subliste de numere întregi, se cere un program SWI-Prolog care returnează o listă care conține toate sublistele care rezultă din concatenarea a câte o sublistă din fiecare listă. De exemplu, pentru următoarele 2 liste [1,2, [4,2], 6, [3,2]] și [1,2,3,[5,6],8, 5,[2,3], 4,1,[3,3]] rezultatul va fi (nu neapărat în această ordine): [[4,2,5,6], [4,2,2,3], [4,2,3,3], [3,2,5,6], [3,2,2,3], [3,2,3,3]].

- C. Să se scrie un program PROLOG care generează lista aranjamentelor de **k** elemente dintr-o listă de numere întregi, având o sumă **S** dată. Se vor scrie modelele matematice și modelele de flux pentru predicatele folosite.

Exemplu- pentru lista [6, 5, 3, 4], **k**=2 și **S**=9 \Rightarrow [[6,3],[3,6],[5,4],[4,5]] (nu neapărat în această ordine)

- D. Se consideră o listă neliniară. Să se scrie o funcție LISP care să aibă ca rezultat lista inițială din care au fost eliminate toate aparițiile unui element **e**. **Se va folosi o funcție MAP.**

Exemplu

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