

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 în LISP

```
(DEFUN F(L)
  (COND
    ((NULL L) NIL)
    ((LISTP (CAR L)) (APPEND (F (CAR L)) (F (CDR L)) (CAR (F (CAR L)))))
    (T (LIST(CAR 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 o listă liniară de numere, se cere un program SWI-Prolog care aplică o sortare stabilă pe această listă și ordonează elementele crescător pe baza restului împărțirii la 3. De exemplu, pentru lista [10, 5, 6, 12, 7, 3, 20, 30] rezultatul va fi [6, 12, 3, 30, 10, 7, 5, 20]. (Obs: sortare stabilă înseamnă că elementele *egale* vor rămâne în aceeași ordine în care erau în lista inițială, de exemplu 6 și 12).

- 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 ≤ 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. 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)))