

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)) 2) (+ (F (CDR L)) (F(CAR L))))
    (T (+ (F (CAR L)) 1))
  )
)
```

Rescrieți această definiție pentru a evita apelul recursiv repetat **(F (CAR L))**. 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 inversează acele subliste în care cel mai mic multiplu comun al elementelor este mai mare decât pătratul primului element al sublistei. **De exemplu,** pentru lista `[[4, 1, 18], 7, 2, -3, [6, 9, 11, 3], 4, [9, 4, 3]]`, rezultatul corect este: `[[18, 1, 4], 7, 2, -3, [3, 11, 9, 6], 4, [9, 4, 3]]`.

C. Să se scrie un program PROLOG care generează lista submulțimilor de sumă **S** dată, cu elementele unei liste, astfel încât numărul elementelor pare din submulțime să fie par. **Exemplu** pentru lista [1, 2, 3, 4, 5, 6, 10] și **S**=10 \Rightarrow [[1,2,3,4], [4,6]].

- D. Se consideră o listă neliniară. Să se scrie o funcție care să aibă ca rezultat lista inițială în care atomii de pe nivelul **k** au fost înlocuiți cu 0 (nivelul superficial se consideră 1). **Se va folosi o funcție MAP.**

Exemplu pentru lista (a (1 (2 b)) (c (d)))

a) k=2 => (a (0 (2 b)) (0 (d))) **b)** k=1 => (0 (1 (2 b)) (c (d))) **c)** k=4 => lista nu se modifică