

Programare logică și funcțională

- examen scris -

Notă

1. Subiectele se notează astfel: of - 1p; A – 2p; B - 4p; C - 3p.
2. Problema Prolog (B) vor fi rezolvată 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. Problema Lisp (C) va fi rezolvată 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(G L)
  (COND
    ((NULL L) NIL)
    (> (FUNCALL G L) 0) (CONS (FUNCALL G L) (F (CDR L))))
    (T (FUNCALL G L))
  )
)
```

Rescrieți această definiție pentru a evita apelul repetat **(FUNCALL G L)**, fără a redefini logica clauzelor și fără a folosi o funcție auxiliară. Nu folosiți SET, SETQ, SETF. Justificați răspunsul.

B. Scrieți un program PROLOG care determină dintr-o listă formată din numere întregi lista subșirurilor cu cel puțin 2 elemente, formate din elemente în ordine strict crescătoare. Se vor scrie modelele matematice și modelele de flux pentru predicatele folosite. ***Exemplu***- pentru lista [1, 8, 6, 4] \Rightarrow [[1,8],[1,6],[1,4],[6,8],[4,8],[4,6],[1,4,6],[1,4,8],[1,6,8],[4,6,8],[1,4,6,8]] (nu neapărat în această ordine)

C. Se consideră o listă neliniară. Să se scrie o funcție LISP care să aibă ca rezultat lista inițială în care toate aparițiile unui element **e** au fost înlocuite cu o valoare **e1**. **Se va folosi o funcție MAP.**

Exemplu

- a)** dacă lista este (1 (2 A (3 A)) (A)) **e** este A și **e1** este B => (1 (2 B (3 B)) (B))
b) dacă lista este (1 (2 (3))) și **e** este A => (1 (2 (3)))