

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(N)
  (COND
    ((= N 1) 1)
    ((> (F (- N 1)) 2) (- N 2))
    ((> (F (- N 1)) 1) (F (- N 1)))
    (T (- (F (- N 1)) 1)))
  )
)
```

Rescrieți această definiție pentru a evita apelul repetat **(F (- N 1))**, fără a redefini logica clauzelor și fără a folosi o funcție auxiliară. Nu folosiți SET, SETQ, SETF. Justificați răspunsul.

B. Dându-se o listă formată din numere întregi, să se genereze în PROLOG lista submulțimilor cu număr par de elemente. Se vor scrie modelele matematice și modelele de flux pentru predicatele folosite.

Exemplu- pentru lista $L=[2,3,4] \Rightarrow [[],[2,3],[2,4],[3,4]]$ (nu neapărat în această ordine)

C. Un arbore n-ar se reprezintă în LISP astfel (nod subarbore1 subarbore2). Se cere să se determine calea de la rădăcină către un nod dat. **Se va folosi o funcție MAP.**

Exemplu pentru arborele (a (b (g)) (c (d (e)) (f)))

a) nod=e => (a c d e) **b)** nod=v => ()