

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 care reprezintă o mulțime, se cere un program SWI-Prolog, care returnează toate posibilitățile de a împărți mulțimea în k submulțimi. Cele k submulțimi trebuie să fie disjuncte și fiecare element din mulțimea inițială trebuie să apară într-una dintre submulțimi. De exemplu, pentru mulțimea $[1,2,3]$ și $k = 2$, soluția este (nu neapărat în această ordine): $[[[3, 2], [1]], [[2], [3, 1]], [[3], [2,1]]]$.

- C. Să se scrie un program PROLOG care generează lista submulțimilor cu valori din intervalul $[a, b]$, având număr par de elemente pare și număr impar de elemente impare. Se vor scrie modelele matematice și modelele de flux pentru predicatele folosite.

Exemplu- pentru $a=2$ și $b=4 \Rightarrow [[2,3,4]]$

- D.** Să se substituie un element **e** prin altul **e1** la orice nivel impar al unei liste neliniare. Nivelul superficial se consideră 1. De exemplu, pentru lista (1 d (2 d (d))), **e**=d și **e1**=f rezultă lista (1 f (2 d (f))). **Se va folosi o funcție MAP.**