

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 Fct(F L)
  (COND
    ((NULL L) NIL)
    ((FUNCALL F (CAR L)) (CONS ((FUNCALL F (CAR L))) (Fct F (CDR L))))
    (T NIL)
  )
)
```

Rescrieți această definiție pentru a evita dublul apel recursiv **(FUNCALL 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 înlocuiește secvențele de numere egale cu suma secvenței. Acest proces trebuie repetat până când nu mai sunt elemente consecutive egale în listă. De exemplu, pentru lista [1, 2, 1, 1, 4, 5, 6, 7, 7, 7, 3, 3, 3, 3, 3, 3, 3, 10], rezultatul va fi [1, 8, 5, 6, 42, 10].

- C. 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)

D. Un arbore n-ar se reprezintă în LISP astfel (nod subarbore1 subarbore2)

Se cere să se înlocuiască nodurile de pe nivelul **k** din arbore cu o valoare **e** dată. Nivelul rădăcinii se consideră a fi 0.

Se va folosi o funcție MAP.

Exemplu pentru arborele (a (b (g)) (c (d (e)) (f))) și **e=h**

- a) $k=2 \Rightarrow (a (b (h)) (c (h (e)) (h)))$
- b) $k=4 \Rightarrow (a (b (g)) (c (d (e)) (f)))$