

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. Dându-se o listă formată din numere întregi, să se genereze lista submulțimilor cu **k** elemente numere impare, în progresie aritmetică. Se vor scrie modelele matematice și modelele de flux pentru predicatele folosite.

Exemplu pentru lista $L=[1,5,2,9,3]$ și $k=3 \Rightarrow [[1,5,9],[1,3,5]]$ (nu neapărat în această ordine)

- D. Se consideră o listă neliniară. Să se scrie o funcție care să aibă ca rezultat lista inițială în care atomii de pe nivelurile pare 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))) se obține (a (0 (2 b)) (0 (d)))