Programare logică și funcțională - examen scris -

<u>Notă</u>

- 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)

((> (CAR L) 0)

(COND

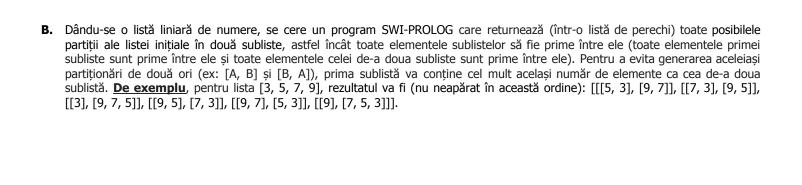
((> (CAR L) (F (CDR L))) (CAR L))

(T (F (CDR L)))

)

(T (F (CDR L)))
```

Rescrieți această definiție pentru a evita apelul recursiv repetat **(F (CDR L))**. Nu redefiniți funcția. Nu folosiți SET, SETQ, SETF. Justificați răspunsul.



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

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

D. Un arbore n-ar se reprezintă în LISP astfel (nod subarbore1 subarbore2). Se cere să se determine lista nodurilor de pe nivelul k. Nivelul rădăcinii se consideră 0. Se va folosi o funcție MAP.

 <u>Exemplu</u> pentru arborele (a (b (g)) (c (d (e)) (f)))
 a) k=2 => (g d)
 b) k=5 => ()