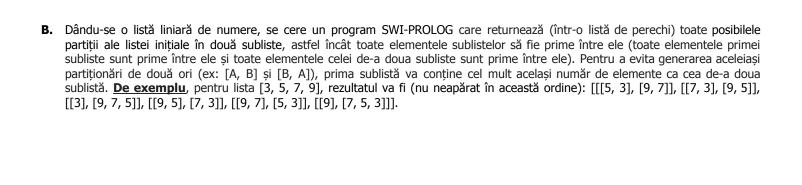
Programare logică și funcțională - examen scris -

<u>Notă</u>

- 1. Subjectele 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(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)). Nu redefiniți funcția. Nu folosiți SET, SETQ, SETF. Justificați răspunsul.



C. Să se scrie un program PROLOG care generează lista aranjamentelor de **k** elemente dintr-o listă de numere întregi, având o sumă **S** dată. Se vor scrie modelele matematice și modelele de flux pentru predicatele folosite.

Exemplu- pentru lista [6, 5, 3, 4], $k=2 \text{ şi } S=9 \Rightarrow [[6,3],[3,6],[5,4],[4,5]]$ (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 => ()