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 predicat PROLOG f(integer, integer), având modelul de flux (i, o):

f(20, -1):-!. f(I,Y):-J is I+1, $\underline{f(J,V)}$, V>0, !, K is J, Y is K. f(I,Y):-J is I+1, $\underline{f(J,V)}$, Y is V-1.

Rescrieți această definiție pentru a evita apelul recursiv **f(J,V)** în ambele clauze. Nu redefiniți predicatul. Justificați răspunsul.



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] şi k=3 \Rightarrow [[1,5,9],[1,3,5]] (nu neapărat în această ordine)

D. Se consideră o listă neliniară. Să se scrie o funcţie LISP care să aibă ca rezultat lista iniţială din care au fost eliminaţi toţi atomii de pe nivelul k (nivelul superficial se consideră 1). Se va folosi o funcţie MAP.

 <u>Exemplu</u> pentru lista (a (1 (2 b)) (c (d)))
 a) k=2 => (a ((2 b)) ((d)))
 b) k=1 => ((1 (2 b)) (c (d)))
 c) k=4 =>lista nu se modifică