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

```
f(0, 0):-!.
f(I,Y):-J is I-1, f(J,V), V>1, !, K is I-2, Y is K.
f(I,Y):-J is I-1, 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.

B. Dându-se o listă neliniară conținând atât atomi numerici, cât și nenumerici, se cere un program LISP care să calculeze numărul total de atomi nenumerici la nivel superficial din acele subliste (incluzând și lista originală) al căror prim atom numeric (la orice nivel) este număr par. **De exemplu**, pentru lista (A B 12 (5 D (A F (10 B) D (5 F) 1)) C 9) rezultatul va fi 7.

C. Să se scrie un program PROLOG care generează lista submulţimilor cu valori din intervalul [**a**, **b**], având număr par de elemente pare și număr impar de elemente impare. Se vor scrie modelele matematice și modelele de flux pentru predicatele folosite.

Exemplu - pentru **a**=2 și **b**=4 ⇒ [[2,3,4]]

D. Se consideră o listă neliniară. Să se scrie o funcţie LISP care să aibă ca rezultat lista iniţială în care toate apariţiile unui element **e** au fost înlocuite cu o valoare **e1**. **Se va folosi o funcție MAP.**

Exemplu **a)** dacă lista este (1 (2 A (3 A)) (A)) **e** este A şi **e1** este B => (1 (2 B (3 B)) (B))

b) dacă lista este (1 (2 (3))) şi **e** este A => (1 (2 (3)))