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 L o listă numerică şi următoarea definiţie de predicat PROLOG având modelul de flux (i, o):

```
f([],-1).
f([H|T],S):-f(T,S1),S1>0,!,S is S1+H.
f([_|T],S):-f(T,S1),S is S1.
```

Rescrieți această definiție pentru a evita apelul recursiv **f(T,S)** 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 construiască o listă care să conțină pentru nivelurile pare cel mai mare atom numeric iar pentru nivelurile impare cel mai mic atom numeric (se presupune că fiecare nivel al listei conține cel puțin un atom numeric), dar în ordine inversă (astfel minimul de pe nivelul 1 este ultimul element, minimul de pe nivelul 2 este penultimul element, etc). **De exemplu**, pentru lista (A B 12 (5 D (A F (10 B) D (5 F) 1)) C 9 (F 4 (D) 9 (F (H 7) K) (P 4)) X) rezultatul va fi (10 1 9 9). Nu este permisă folosirea funcției *reverse* din Lisp.

C. Să se scrie un program PROLOG care generează lista submulţimilor formate cu elemente unei liste listă de numere întregi, având număr suma elementelor număr impar și număr par nenul de elemente pare. Se vor scrie modelele matematice și modelele de flux pentru predicatele folosite.

Exemplu- pentru lista [2,3,4] $\Rightarrow$ [[2,3,4]]

D. Un arbore n-ar se reprezintă în LISP astfel (nod subarbore1 subarbore2 .....)
Se cere să se înlocuiască nodurile de pe nivelurile impare din arbore cu o valoare **e** dată. Nivelul rădăcinii se consideră a fi 0. **Se va folosi o funcție MAP.**

Exemplu pentru arborele (a (b (g)) (c (d (e)) (f))) şi **e**=h => (a (h (g)) (h (d (h)) (h)))