Programare logică și funcțională
- examen scris -

Notă

1. Subiectele se notează astfel: of - 1p; A – 2p; B - 4p; C - 3p.
2. Problema Prolog (B) vor fi rezolvată 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. Problema Lisp (C) va fi rezolvată 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 **f(list, integer)**, având modelul de flux (i, o):

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

Rescrieți această definiție pentru a evita apelul recursiv **f(T,S)** în ambele clauze, fără a redefini logica clauzelor. Justificați răspunsul.

B. 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 şi **S**=9⇒ [[6,3],[3,6],[5,4],[4,5]] (nu neapărat în această ordine)

C. Să se substituie valorile numerice cu o valoare **e** dată, la orice nivel al unei liste neliniare. **Se va folosi o funcție MAP.**
Exemplu, pentru lista (1 d (2 f (3))), **e**=0 rezultă lista (0 d (0 f (0))).