

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

$f(0, 0) :- !.$

$f(I, Y) :- J \text{ is } I-1, \text{ **f(J, V)**, } V > 1, !, K \text{ is } I-2, Y \text{ is } K.$

$f(I, Y) :- J \text{ is } I-1, \text{ **f(J, V)**, } Y \text{ is } V+1.$

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

B. Să se scrie un program PROLOG care generează lista combinațiilor de **k** elemente cu numere de la 1 la **N**, având diferența între două numere consecutive din combinație număr par. Se vor scrie modelele matematice și modelele de flux pentru predicatele folosite. **Exemplu** pentru **N=4**, **k=2** \Rightarrow $[[1,3],[2,4]]$ (nu neapărat în această ordine)

C. Se dă o listă neliniară și se cere înlocuirea valorilor numerice pare cu numărul natural succesor. **Se va folosi o funcție MAP.**

Exemplu pentru lista (1 s 4 (2 f (7))) va rezulta (1 s 5 (3 f (7))).