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 funcţie LISP

```
(DEFUN F(N)
        (COND
                ((= N 1) 1)
                (> (F (- N 1)) 2) (- N 2))
                (> (F (- N 1)) 1) (F (- N 1)))
                (T (- (F (- N 1)) 1))
        )
)
```

Rescrieți această definiție pentru a evita apelul repetat **(F (- N 1))**. Nu redefiniți funcția. Nu folosiți SET, SETQ, SETF. Justificați răspunsul.

B. Dându-se 2 liste formate din numere întregi și subliste de numere întregi, se cere un program SWI-Prolog care returnează o listă care conține, pentru fiecare pereche posibilă de subliste (o sublistă din prima listă și una din a doua), produsul elementelor maxime. De exemplu, pentru următoarele 2 subliste [1,2, [4,2], 6, [3,2]] și [1,2,3,[5,6],8, 5,[12,3], 4,1,[3,8]] rezultatul va fi (nu neapărat în această ordine): [24, 48, 32, 18, 36, 24].

C. Să se scrie un program PROLOG care generează lista submulţimilor de sumă pară, cu elementele unei liste. Se vor scrie modelele matematice și modelele de flux pentru predicatele folosite.

Exemplu- pentru lista L=[2, 3, 4] ⇒ [[],[2],[4],[2,4]] (nu neapărat în această ordine)

D. 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))).