

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 definiție de funcție LISP

```
(DEFUN F(L)
  (COND
    ((NULL L) 0)
    ((> (F (CAR L)) 2) (+ (CAR L) (F (CDR L))))
    (T (F (CAR L))))
  )
)
```

Rescrieți această definiție pentru a evita dublul apel recursiv (F (CAR L)). Nu redefiniți funcția. Nu folosiți SET, SETQ, SETF. Justificați răspunsul.

- B.** Dându-se o listă liniară de numere, se cere un program SWI-Prolog care aplică o sortare stabilă pe această listă și ordonează elementele crescător pe baza restului împărțirii la 3. De exemplu, pentru lista [10, 5, 6, 12, 7, 3, 20, 30] rezultatul va fi [6, 12, 3, 30, 10, 7, 5, 20]. (Obs: sortare stabilă înseamnă că elementele *egale* vor rămâne în aceeași ordine în care erau în lista inițială, de exemplu 6 și 12).

- C. Dându-se o listă formată din numere întregi, să se genereze în PROLOG lista submulțimilor cu număr par de elemente. Se vor scrie modelele matematice și modelele de flux pentru predicatele folosite.

Exemplu pentru lista $L=[2,3,4] \Rightarrow [[],[2,3],[2,4],[3,4]]$ (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))).