

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 în LISP

```
(DEFUN F(L)
  (COND
    ((ATOM L) -1)
    ((> (F (CAR L)) 0) (+ (CAR L) (F (CAR L)) (F (CDR L))))
    (T (F (CDR L)))
  )
)
```

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

- B.** Pentru 2 liste care reprezintă numere cifră cu cifră, definim operația de *rearanjare* în modul următor: comparăm cifrele și formăm 2 liste noi, una conținând cifrele mai mari și una cu cifrele mai mici. De exemplu, rearanjarea listelor [1,4,2,7] și [8,2,9,6,1] va returna [8, 2, 9, 6, 7] și [1, 4, 2, 1]. Dându-se 2 liste care conțin numere și subliste cu cifre, se cere un program SWI-Prolog care returnează 2 liste cu subliste, ele conținând rezultatul după rearanjarea sublistelor din cele 2 liste. Una dintre liste conține sublistele cu cifrele mai mari și cealaltă conține sublistele cu cifrele mai mici. Cele două liste de intrare au același număr de subliste, dar nu neapărat pe aceeași poziții. De exemplu, pentru listele [1, 2, [6, 2, 4], 6, [9, 9, 1, 1], 17, 9, [5, 3, 8, 1, 9]] și [1, 2, 3, [1, 5], 7, 11, [8, 3], 7, 5, [9, 4, 2, 5], 77] rezultatul va fi [[6, 2, 5], [9, 9, 8, 3], [5, 9, 8, 2, 9]] și [[1, 4], [1, 1], [3, 4, 1, 5]].

C. Să se scrie un program PROLOG care generează lista submulțimilor de sumă **S** dată, cu elementele unei liste, astfel încât numărul elementelor pare din submulțime să fie par. **Exemplu** pentru lista [1, 2, 3, 4, 5, 6, 10] și **S**=10 \Rightarrow [[1,2,3,4], [4,6]].

- D.** Să se substituie un element **e** prin altul **e1** la orice nivel impar al unei liste neliniare. Nivelul superficial se consideră 1. De exemplu, pentru lista (1 d (2 d (d))), **e**=d și **e1**=f rezultă lista (1 f (2 d (f))). **Se va folosi o funcție MAP.**