

# 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 **G** o funcție LISP și fie următoarea definiție

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

Rescrieți această definiție pentru a evita apelul repetat (**G L**). Nu redefiniți funcția. Nu folosiți SET, SETQ, SETF. Justificați răspunsul.

- B.** Dându-se o listă care reprezintă o mulțime, se cere un program SWI-Prolog, care returnează toate posibilitățile de a împărți mulțimea în  $k$  submulțimi. Cele  $k$  submulțimi trebuie să fie disjuncte și fiecare element din mulțimea inițială trebuie să apară într-una dintre submulțimi. De exemplu, pentru mulțimea  $[1,2,3]$  și  $k = 2$ , soluția este (nu neapărat în această ordine):  $[[[3, 2], [1]], [[2], [3, 1]], [[3], [2,1]]]$ .

- C. Să se scrie un program PROLOG care generează lista combinărilor de **k** elemente dintr-o listă de numere întregi, având suma număr par. Se vor scrie modelele matematice și modelele de flux pentru predicatele folosite.

**Exemplu** pentru lista [6, 5, 3, 4], **k**=2  $\Rightarrow$  [[6,4],[5,3]] (nu neapărat în această ordine)

- D. Se dă o listă neliniară și se cere înlocuirea valorilor numerice care sunt mai mari decât o valoare **k** dată și sunt situate pe un nivel impar, cu numărul natural predecesor. Nivelul superficial se consideră 1. **Se va folosi o funcție MAP.**

**Exemplu** pentru lista (1 s 4 (3 f (7))) și

a)  $k=0$  va rezulta (0 s 3 (3 f (6)))

b)  $k=8$  va rezulta (1 s 4 (3 f (7)))