**L1** Programare recursivă ȋn Lisp (I)

**Data predare**: 15.12.2021

**Numarul + enuntul problemei:** 6

**a)** Sa se scrie de doua ori elementul de pe pozitia a n-a a unei liste liniare. De exemplu, pentru (10 20 30 40 50) si n=3 se va produce (10 20 30 30 40 50).

**b)** Sa se scrie o functie care realizeaza o lista de asociere cu cele doua liste pe care le primeste. De ex: (A B C) (X Y Z) --> ((A.X) (B.Y) (C.Z)).

**c)** Sa se determine numarul tuturor sublistelor unei liste date, pe orice nivel. Prin sublista se intelege fie lista insasi, fie un element de pe orice nivel, care este lista. Exemplu: (1 2 (3 (4 5) (6 7)) 8 (9 10)) => 5 (lista insasi, (3 ...), (4 5), (6 7), (9 10)).

**d)** Sa se construiasca o functie care intoarce numarul atomilor dintr-o lista, de la nivel superficial.

**Modelele recursive:**

**a)**

dublare\_el\_poz(l1…ln, poz, n) = { vida, daca l1…ln e vida

{ l1 U l1 U dublare\_el\_poz(l2…ln, poz+1, n), daca poz=n

{ l1 U dublare\_el\_poz(l2…ln, poz+1, n), daca poz!=n

dublareAux(l1…ln, n) = dublare\_el\_poz(l1…ln, 1 , n)

**b)**

pereche(l1…ln, k1…kn) = { vida, daca l1 e vida sau k1 e vida

{ (l1 U k1) U pereche(l2…ln, k2…kn)

**c)**

nrListe(l1…ln) = { 1, daca l1…ln e vida

{ nrListe(l1) + nrListe(l2…ln), daca l1 lista

{ nrListe(l2…ln), daca l1 nu e lista

**d)**

nrAtomi(l1…ln) = { 0, daca l1…ln e vida

{ nrAtomi(l2…ln), daca l1 e lista

{ 1 + nrAtomi(l2…ln), daca l1 e atom

**Cod sursa:**

**Exemple de testare:**