



1. Să se scrie o funcție care întoarce adâncimea unei liste.

Exemplu:  $((1\ 2)\ 3\ (4\ (5\ 6))) \rightarrow$  adâncimea e 3

Model recursiv

$$\text{adâncime}(x) = \begin{cases} 0, & \text{dacă } x \text{ e atom} \\ 1 + \max(\text{adâncime}(x_1), \text{adâncime}(x_2), \dots, \text{adâncime}(x_m)), & \text{dacă } x \text{ e listă} \end{cases}$$

3. Să se construiască o funcție care verifică dacă un atom e membru al unei liste nu depășind limita.

Model recursiv

$$\text{membru}(x, e) = \begin{cases} 0, & \text{dacă } x \text{ e atom și } x \neq e \\ 1, & \text{dacă } x \text{ e atom și } x = e \\ \sum_{i=1}^n \text{membru}(x_i), & \text{altfel} \end{cases}$$

$$\text{main}(x, e) = \begin{cases} \text{"e membru"}, & \text{dacă } \text{membru}(x, e) \neq 0 \\ \text{"Nu e membru"}, & \text{dacă } \text{membru}(x, e) = 0 \end{cases}$$

4. Să se construiască o funcție care întoarce suma atomilor numerici dintr-o listă, la orice nivel.

Model recursiv

$$\text{suma}(x) = \begin{cases} x, & \text{dacă } x \text{ e atom numeric} \\ 0, & \text{dacă } x \text{ e atom nenumeric} \\ \sum_{i=1}^n \text{suma}(x_i), & \text{dacă } x \text{ e listă} \end{cases}$$

5. Definiți o funcție care testează apartenența unui mod într-un arbore n-or reprezentat sub formă (rezolvăm listă-moduri-subarb, ... listă-moduri-subarb<sub>m</sub>).

Exemplu: arborele este  $(a\ (b\ (c))\ (d)\ (e\ (f)))$  și modul este 'b'  $\Rightarrow$  aderă

Model recursiv

$$\text{mod}(x, e) = \begin{cases} 1, & \text{dacă } x = e \text{ și } x \text{ atom} \\ 0, & \text{dacă } x \neq e \text{ și } x \text{ atom} \\ \sum_{i=1}^n (\text{mod}(x_i)), & \text{altfel} \end{cases}$$

$$\text{main}(x, e) = \begin{cases} \text{"aderă"}, & \text{dacă } \text{mod}(x, e) \neq 0 \\ \text{"fals"}, & \text{altfel} \end{cases}$$

6. Să se construiască o funcție care calculează produsul atomilor numerici dintr-o listă, la orice nivel.

Model recursiv

$$\text{produs}(x) = \begin{cases} x, & \text{dacă } x \text{ e atom numeric} \\ 1, & \text{dacă } x \text{ e atom nenumeric} \\ \prod_{i=1}^n (\text{produs}(x_i)), & \text{altfel} \end{cases}$$

8. Să se construiască o funcție care întoarce maximul atomilor numerici dintr-o listă, de la orice nivel.

Model recursiv

$$\text{maxim}(x) = \begin{cases} x, & \text{dacă } x \text{ e atom numeric} \\ -1, & \text{dacă } x \text{ e atom nenumeric} \\ \max(\text{produs}(x_i)), & \text{altfel} \end{cases}$$

9. Definiți o funcție care substituie un element  $\in$  prin elementele unei liste  $L$ , la toate nivelurile unei date  $L$ .

Model recursiv

$$\text{substituie-listă}(x, e, l) = \begin{cases} x, & \text{dacă } x \neq e \text{ și } x \text{ atom} \\ e, & \text{dacă } x = e \text{ și } x \text{ atom} \\ \bigcup_{i=1}^n \text{substituie-listă}(x_i, e, l), & \text{altfel} \end{cases}$$

10. Definiți o funcție care determină numărul nodurilor de pe nivelul  $k$  dintr-un arbore  $n$ -ar reprezentat sub formă (rădăcină listă-noduri-subarb, ... listă-noduri-subarb $_n$ ).

Exemplu:  $(a(b(c))(d)(e(f)))$  și  $k=1 \Rightarrow 3$  noduri

Model recursiv

$$\text{nr-noduri}(l, \text{niv}, k) = \begin{cases} 1, & \text{dacă } x = \text{atom și } \text{niv} = k \\ 0, & \text{dacă } x = \text{atom și } \text{niv} \neq k \\ \sum_{i=1}^n (\text{nr-noduri}(x_i, k, \text{niv}+1)), & \text{altfel} \end{cases}$$

11. Să se scrie o funcție care șterge toate aparițiile unui atom de la toate nivelurile unei liste.

Model recursiv

$$\text{șterge}(x, e) = \begin{cases} x, & \text{dacă } x \text{ atom și } x \neq e \\ \text{nil}, & \text{dacă } x \text{ atom și } x = e \\ \bigcup_{i=1}^n \text{șterge}(x_i, e), & \text{altfel} \end{cases}$$

12. Definiți o funcție care înlocuiește un nod cu altul într-un arbore  $n$ -ar reprezentat sub formă de (rădăcină listă-noduri-subarb, ... listă-noduri-subarb $_n$ ).

Exemplu:  $(a(b(c))(d)(e(f)))$  și nodul  $b$  cu nodul  $g \Rightarrow (a(g(c))(d)(e(f)))$

Model recursiv

$$\text{înlocuire}(x, e, f) = \begin{cases} f, & \text{dacă } x \text{ e atom și } x = e \\ x, & \text{dacă } x \text{ e atom și } x \neq e \\ \bigcup_{i=1}^n \text{înlocuire}(x_i, e, f), & \text{altfel} \end{cases}$$

13. Definiți o funcție care substituie un element prin altul la toate nivelurile unei liste date.

Model recursiv

$$\text{înlocuire}(x, e, f) = \begin{cases} f, & \text{dacă } x \text{ e atom și } x = e \\ x, & \text{dacă } x \text{ e atom și } x \neq e \\ \bigcup_{i=1}^n \text{înlocuire}(x_i, e, f), & \text{altfel} \end{cases}$$

14. Definiți o funcție care dă adâncimea unui arbore  $n$ -ar reprezentat sub formă (rădăcină listă-noduri-subarb, ... listă-noduri-subarb $_n$ ).

Exemplu:  $(a(b(c))(d)(e(f)))$  este 3

Model recursiv

$$\text{adâncime}(x) = \begin{cases} 0, & \text{dacă } x \text{ e atom} \\ 1 + \max_{i=1}^n (\text{adâncime}(x_i)), & \text{altfel} \end{cases}$$

15. Să se construiască o funcție care întoarce numărul atomilor dintr-o listă, de la orice nivel.

Model recursiv

$$\text{nr-atomi}(x) = \begin{cases} 1, & \text{dacă } x = \text{atom} \\ \sum_{i=1}^n \text{nr-atomi}(x_i), & \text{altfel} \end{cases}$$

16. Definim o funcție care inversează o listă împreună cu toate sublistele sale de pe orice nivel.

Model recursiv

$$\text{inversează}(x) = \begin{cases} x, & \text{dacă } x = \text{atom} \\ \sum_{i=1}^n \text{inversează}(x_i), & \text{altfel} \end{cases}$$

7. Să se scrie o funcție care calculează suma numerelor pare minus suma numerelor impare la toate nivelurile unei liste.

Model recursiv

$$\text{sumă-pare-impare}(x) = \begin{cases} x, & \text{dacă } x \text{ e atom și e și par} \\ -x, & \text{dacă } x \text{ e atom și e și impar} \\ \sum_{i=1}^n \text{sumă-pare-impare}(x_i), & \text{altfel} \end{cases}$$