

Recapitulare pentru colocviu

1. Calculați cel mai mare divizor comun a două numere.
Ex: ? - *cmmdc*(15,25,R).
R = 5.
2. Calculați cel mai mic multiplu comun a două numere.
Ex: ? - *cmmmc*(15,25,R).
R = 75.
3. Calculați suma elementelor unei liste.
Ex: ? - *suma*([1,2,3,4,5], R).
R = 15.
4. Extrageți numerele pare în E și numerele impare în O.
Ex: ? - *separate_parity*([1,2,3,4,5,6], E, O).
E = [2,4,6], O=[1,3,5].
5. Înlocuiți toate aparițiile lui X cu Y.
Ex: ? - *replace_all*(1, a, [1,2,3,1,2], R).
R = [a,2,3,a,2].
6. Ștergeți elementele de pe poziții divizibile cu K.
Ex: ? - *sterge_k*([6,5,4,3,2,1], 3, R).
R = [6,5,3,2].
7. Inversează o listă completă/incompletă.
Ex: ? - *reverse*([1, 2, 3, 4, 5], R).
R = [5, 4, 3, 2, 1].
8. Calculați adâncimea maximă a unei liste imbricate.
Ex: ? - *adancime*([1, [2, [3]], [4]], R1), *adancime*([], R2).
R1 = 3, R2 = 1.
9. Aplatizați o listă imbricată cu liste complete/incomplete.
Ex: ? - *flat*([1, [2, [3]], [4]], R).
R = [1,2,3,4].
10. Calculați adâncimea unui arbore binar complet/incomplet.
Ex: *tree*(t(6, t(4, t(2, nil, nil)), t(5, nil, nil)), t(9, t(7, nil, nil), nil))).
? - *tree*(T), *adancime*(T, R).
R = 3.
11. Colectați toate nodurile unui arbore binar complet/incomplet în inordine folosind liste complete/diferență.
Ex: *tree*(t(6, t(4, t(2, nil, nil)), t(5, nil, nil)), t(9, t(7, nil, nil), nil))).
? - *tree*(T), *inordine*(T, R).
R = [2,4,5,6,7,9].
12. Colectați toate frunzele dintr-un arbore binar.
Ex: *tree*(t(6, t(4, t(2, nil, nil)), t(5, nil, nil)), t(9, t(7, nil, nil), nil))).
? - *tree*(T), *collect_k*(T, R).
R = [2,5,7].
13. Ștergeți elementele duplicate dintr-o listă (păstrează prima sau ultima apariție).
Ex: ? - *sterge_duplicate*([3,4,5,3,2,4], R).
R = [3,4,5,2]. sau R = [5,3,2,4].
14. Ștergeți toate aparițiile elementului minim/maxim dintr-o listă.
Ex: ? - *sterge_min*([4,5,1,2], R).
R = [4,5,2].
15. Inversați elementele dintr-o lista după poziția K.
Ex: ? - *reverse_k*([1,2,3,4,5,6], 2, R).
R = [1,2,6,4,5,3].
16. Codificați o listă cu RLE (*Run-length encoding*). Doua sau mai multe elementele consecutive se înlocuiesc cu (element, nr_apariții). Dacă nr_apariții=1 atunci se scrie doar elementul.
Ex: ? - *rle_encode*([1,1,1,2,3,3,4,4], R).
R = [(1,3), 2, (3,2), (4,2)].

17. Rotiți lista K poziții în dreapta.
Ex: ? – rotate_k([1,2,3,4,5,6], 2, R).
R = [5,6,1,2,3,4].
18. Sortați o listă de caractere în funcție de codul ASCII.
Ex: ? – sort_chars([e, t, a, v, f], R).
R = [a, e, f, t, v].
19. Sortați o listă de liste în funcție de lungimea listelor de nivel 2.
Ex: ? – sort_len([[a, b, c], [f], [2, 3, 1, 2], [], [4, 4]], R).
R = [[], [f], [4, 4], [a, b, c], [2, 3, 1, 2]].
20. Calculați suma elementelor de la nivelul K într-o lista imbricată.
Ex: ? – sum_k([1, [2, [3]], [4]], 2, R).
R = 6.
21. Înlocuiți toate aparițiile lui X cu Y în lista imbricată.
Ex: ? – replace_all(2, 5, ([1, [2, [3, 2]], [4]], R).
R = [1, [5, [3, 5]], [4]].
22. Colectați toate nodurile de la adâncimea K dintr-un arbore binar.
Ex: tree(t(6, t(4, t(2, nil, nil)), t(5, nil, nil)), t(9, t(7, nil, nil), nil)).
? – tree(T, collect_k(T, 2, R).
R = [4, 9].
23. Calculați gradul interior/exterior al unui nod în graf.
Ex: nod(1). nod(2). nod(3). arc(2,1). arc (1,2). arc(3,1).
? – grad_interior(1,R).
R = 2. (arcul 2 -> 1 și arcul 3 -> 1)
24. Colectați toate nodurile unui graf.
Ex: nod(1). nod(2). nod(3).
? – collect(R).
R = [1,2,3].

Studiați:

- operații aritmetice
- operații pe liste (complete, imbricate, diferență, incomplete)
- 1 algoritm de sortare
- operații pe arbori (compleți, incompleți)
- grafuri și efecte laterale (assert, retract)