

1. Se da o lista de $n-1$ numere intregi si aceste numere întregi sunt în intervalul $[1, n]$. Lungimea listei reprezinta valoarea lui $n - 1$. Nu exista duplicate in lista. In lista lipseste unul dintre numerele intregi din intervalul $[1, n]$. Scrieti o functie iterativa `missingNumber` care primeste aceasta lista ca si argument si returneaza numarul lipsa. Nu este necesara folosirea unei liste de asocieri/proprietati.
! Complexitatea trebuie sa fie $O(n^2)$. !

ex: `(missingNumber '(1 3 4 5 6)) ----> 2`
`(missingNumber '(1 2 3 4 5)) ----> 6`

2. Se da o lista formata din trei numere naturale, care reprezinta lungimile unui triunghi dreptunghic. Ordinea lungimilor este arbitrara. Definiti o functie `getIpot` care returneaza lungimea ipotenuzei.

ex: `(getIpot '(3 5 4)) ----> 5`
`(getIpot '(3 4 5)) ----> 5`

3. Se da o lista cu doua numere intregi a si b , care reprezinta coeficientii ecuatiei de gradul I cu o singura necunoscuta ($ax + b = 0$). Definiti o functie `getSolution` care returneaza valoarea necunoscutei x (nu sub forma de fractie / numar rational).

ex: `(getSolution '(1 2)) ----> -2`
`(getSolution '(-4 -2)) ----> -0.5`

4. Se da un numar natural n . Sa se scrie o functie iterativa `getProdDigitsIter` care returneaza produsul cifrelor lui n .

ex `(getSumDigitsIter 0) → 0`
`(getSumDigitsIter 1280) → 11`

5. Se da o lista de $n-1$ numere intregi si aceste numere întregi sunt în intervalul $[1, n]$. Lungimea listei reprezinta valoarea lui $n - 1$. Nu exista duplicate in lista. In lista lipseste unul dintre numerele intregi din intervalul $[1, n]$. Scrieti o functie iterativa `missingNumberOptim` care primeste aceasta lista ca si argument si returneaza numarul lipsa. Nu este necesara folosirea unei liste de asocieri/proprietati.
! Complexitatea trebuie sa fie $O(n)$. !

ex: `(missingNumberOptim '(1 3 4 5 6)) ----> 2`
`(missingNumberOptim '(1 2 3 4 5)) ----> 6`

6. Se dau doua numere intregi a si b. Definiti o functie iterativa rangeListEvenIter care returneaza o lista cu toate numerele întregi pare din intervalul [a, b]. Daca $a > b$, atunci intervalul devine [b, a].

ex: (rangeListEvenIter 4 9) ----> (4 6 8)
(rangeListEvenIter 11 -3) ----> (-2 0 2 4 6 8 10)

7. Se dau doua numere intregi a si b. Definiti o functie recursiva rangeListEvenRec care returneaza o lista cu toate numerele întregi pare din intervalul [a, b]. Daca $a > b$, atunci intervalul devine [b, a].

ex: (rangeListEvenRec 4 9) ----> (4 6 8)
(rangeListEvenRec 11 -3) ----> (-2 0 2 4 6 8 10)

8. Se da o lista formata din trei numere intregi. Definiti o functie checkTriangle care verifica daca cele trei valori pot constitui lungimile laturilor unui triunghi. In caz afirmativ, se returneaza "da", altfel "nu".

ex: (checkTriangle '(1 2 3)) ----> "nu"
(checkTriangle '(2 3 4)) ----> "da"
(checkTriangle '(2 -3 5)) ----> "nu"

9. Se da o lista formata din trei numere naturale. Definiti o functie checkTriangleSquare care verifica daca cele trei valori pot constitui lungimile laturilor unui triunghi dreptunghic. In caz afirmativ, se returneaza "da", altfel "nu".

ex: (checkTriangleSquare '(3 4 5)) ----> "da"
(checkTriangleSquare '(2 2 2)) ----> "nu"

10. Se da o lista formata din trei numere naturale, care reprezinta lungimile unui triunghi. Tipul triunghiului nu este important. Definiti o functie recursiva getSemiperimetru care calculeaza si returneaza semiperimetrul triunghiului. Diferentele de precizie sunt acceptate.

ex: (getSemiperimetru '(2 3 4)) ----> 4.5

11. Se da un numar x si urmatoarele functii $f(x)=4x-1$, $g(x)=\sqrt{6+7x}$. Definiti o functie getFunctionResults care returneaza o lista formata din rezultatele lui f(x), g(x), $(f \circ g)(x)$ si $(g \circ f)(x)$. Diferentele de precizie sunt acceptate.

ex: (getFunctionResults 1) ----> (3 3.605 13.422 5.196)

12.

Se dau doua numere naturale n si k , unde $0 \leq k \leq n < 10$. Definiti o functie `computeCAndA` care returneaza o lista formata din numarul combinarilor de n elemente luate cate k fara repetitii, respectiv numarul aranjamentelor de n elemente luate cate k fara repetitii.

ex (computeCAndA 6 3) \rightarrow (20 120)

13. Se da o lista de $n-1$ numere intregi si aceste numere întregi sunt în intervalul $[1, n]$. Lungimea listei reprezinta valoarea lui $n - 1$. Nu exista duplicate in lista. In lista lipseste unul dintre numerele intregi din intervalul $[1, n]$. Scrieti o functie recursiva `missingNumberRec` care primeste aceasta lista ca si argument si returneaza numarul lipsa. Nu este necesara folosirea unei liste de asocieri/proprietati.
! Complexitatea trebuie sa fie $O(n^2)$. !

ex: (missingNumberRec '(1 3 4 5 6)) ----> 2
(missingNumberRec '(1 2 3 4 5)) ----> 6

14. Se da o lista formata din n numere naturale. In lista exista un singur numar duplicat. Definiti o functie recursiva `getDuplicateRec` care returneaza acest numar duplicat.

ex: (getDuplicateRec '(4 5 3 8 11 2 0 2)) ---> 2

15. Se da un numar natural n . Sa se scrie o functie recursiva `getProdDigitsRec` care returneaza produsul cifrelor lui n .

ex (getProdDigitsRec 0) \rightarrow 0
(getProdDigitsRec 1280) \rightarrow 11

Ich 12:03

1. Ovidiu Santa
2. Alexandru Morar
3. Andrei Oprica

Ich 12:05

- 4 Mihai Olaf
5. Timea Nemeth
6. Benjamin Miklo
7. Diana Trif
8. Florin Marut

Ich 12:06

9. Lorena Pinte
- 10 Georgiana Marunt
11. Marian Bogdan

Ich 12:07

12. Andrei Popescu
13. Marina Miron
14. Robert Sabau
15. Radu Tiberiu

Nachricht an alle senden

