Probleme culese de pe internet.

01.

Pentru fiecare dintre cele 6 cești de cafea pe care le cumpăr, primesc o a 7-a ceașcă gratuită. În total, primesc deci 7 cești. Creați o funcție care preia `n` pentru numărul de cești cumpărate și returnați numărul total de cești pe care le-aș primi.

Exemple

cups(6) ⟶ 7

cups(12) ⟶ 14

cups(213) ⟶ 248

Note

- Returnul este numărul de căni pe care le-am cumpărat + numărul de căni pe care le-am primit gratuit.
- Funcția va primi ca și argument valoarea parametrului `n`. Introducerea de la consolă a lui `n` și validarea acestuia ca fiind număr întreg pozitiv se va face în afara funcției, înainte de apelarea ei. Apelarea se va face având ca și argument numărul citit și validat.

02.

Creați o funcție care ia un număr `num` și returnează primii 10 multipli ai `num` cu 1 adăugat, despărțiți prin virgule.

Exemple

nPlusOne(7) ⟶ [8,15,22,29,36,43,50,57,64,71]

nPlusOne (1) ⟶ [2,3,4,5,6,7,8,9,10,11]

nPlusOne 3) ⟶ [4,7,10,13,16,19,22,25,28,31]

Note

- Nu există virgulă după ultimul număr.
- Funcția va primi ca și argument valoarea parametrului `num`. Introducerea de la consolă a lui `num` și validarea acestuia ca fiind număr întreg pozitiv se va face în afara funcției, înainte de apelarea ei. Apelarea se va face având ca și argument numărul citit și validat.

03.

Este dat un număr întreg `n`. Sarcina ta este să găsești câte cifre conține acest întreg fără a folosi metodele `str` sau `len` !

Exemple

digits(100) ⟶ 3

digits(1000) ⟶ 4

Note

- Funcția va primi ca și argument valoarea parametrului `n`. Introducerea de la consolă a lui `n` și validarea acestuia ca fiind număr întreg pozitiv se va face în afara funcției, înainte de apelarea ei. Apelarea se va face având ca și argument numărul citit și validat.

04.

Creați o funcție care primește doi parametri, o listă de numere `lst` și un număr întreg mai mare sau cel puțin egal cu zero , `n` , și returnează cele **n numere cele mai mari** din lista dată. Argumentele funcției se citesc și se validează înainte de apelarea acesteia.

Exemple

largestNumbers(2, [4, 3, 2, 1]) ➞ [3, 4]

largestNumbers(1, [7, 19, 4, 2]) ➞ [19]

largestNumbers(3, [14, 12, 57, 11, 18, 16]) ➞ [16, 18, 57]

largestNumbers(0, [1, 3, 4, 2]) ➞ []

Note

Lista returnată trebuie sortată în ordine crescătoare.

05.

Sam și Frodo trebuie să fie apropiați. Dacă sunt unul lângă altul în listă, funcția ta ar trebui să returneze `True`. Dacă există un nume între ele, returnează `False`.

Exemple

middleEarth(["Frodo", "Sam", "Gandalf"]) ➞ True

middleEarth(["Frodo", "Saruman", "Sam"]) ➞ False

middleEarth(["Orc", "Sam", "Frodo", "Legolas"]) ➞ True

Note

- Indiferent cine vine primul, rezultatul trebuie să fie `True` dacă Frodo și Sam sunt unul lângă altul.
- Există doar un Sam și un Frodo în listă.
- Lista se citește de la consolă înainte de apelarea funcției.

06.

O valoare este **omniprezentă** dacă există în fiecare sublistă din lista principală.

Pentru a ilustra:

```
[[3, 4], [8, 3, 2], [3], [9, 3], [5, 3], [4, 3]]
# valoarea 3 există în fiecare sublistă, deci este omniprezentă.
```

Creați o funcție care determină dacă o valoare de intrare este **omniprezentă** pentru o listă dată.

Exemple

omniprezent([[1, 1], [1, 3], [5, 1], [6, 1]], 1) ⟶ True

omniprezent ([[1, 1], [1, 3], [5, 1], [6, 1]], 6) ⟶ False

omniprezent ([[5], [5], [5], [6, 5]], 5) ⟶ True

omniprezent ([[5], [5], [5], [6, 5]], 6) ⟶ False

Note

Sub-listele pot avea orice lungime.

07.

Creați o funcție pentru a multiplica toate valorile dintr-o listă cu numărul de valori din lista dată.

Exemple

multiplicare([2, 3, 1, 0]) ⟶ [8, 12, 4, 0]

multiplicare ([4, 1, 1]) ⟶ ([12, 3, 3])

multiplicare ([1, 0, 3, 3, 7, 2, 1]) ⟶ [7, 0, 21, 21, 49, 14, 7]

multiplicare ([0]) ⟶ ([0])

Note

- Toate valorile date sunt numere.
- Toate listele vor avea cel puțin un element.
- Nu uitați `return` de rezultat.

08.

Creați o funcție care preia o listă `lst` și returnează tipurile de valori (tipuri de date) într-o listă nouă.

Exemple

types([1, 10]) ⟶ ['int', 'int'])

types([["hello" , 1] , 10]) ⟶ ['list', 'int'])

types(["shashwat", 10, 90]) ⟶ ['str', 'int', 'int'])

09.

Creați o funcție care ia un număr ca și argument. Adunați toate numerele de la 1 la numărul pe care l-ați transmis funcției. De exemplu, dacă intrarea este 4, atunci funcția ar trebui să returneze 10 deoarece 1 + 2 + 3 + 4 = 10.

Exemple

aduna(4) ⟶ 10

aduna (13) ⟶ 91

aduna (600) ⟶ 180300

Note

Este permis ca și argument orice număr întreg pozitiv între 1 și 1000.

Reguli!!!!!

Puteți utiliza orice căutare pe internet pentru a consulta noțiuni teoretice referitoare la problemele din temă (inclusiv ChatGPT)

Nu puteți utiliza soluții totale sau parțiale de pe internet. Orice parte a soluției trebuie să fie creație proprie.