



Romanian



Test Restanta/Marire - Scor Final: Puncte

Aveti voie DOAR cu documentatia oficiala Python si pdf-urile de la curs. Nu aveti voie sa folositi eval.
Inainte de a incepe testul, asigurati-va ca sunt inchise toate tab-urile din browser si nu exista proiecte deschise in editor.
Orice tentativa de frauda atrage masurile de rigoare corespunzatoare.

Sa se faca un modul python (numele fisierului - *restanta.py*)
In acest modul se vor scrie rezolvarile pentru toate problemele de la acest test.

[Date de testare](#)

[Fisiere de testare](#)

Barem

- Problema 0 = 5 points
- Problema 1 = 12 points
- Problema 2 = 13 points
- Problema 3 = 5 points
- Problema 4 = 7 points
- Problema 5 = 10 points
- Problema 6 = 9 points
- Problema 7 = 9 points

Total = 70 points

Problema 0 - Total: Puncte

Să se scrie o funcție cu numele **problema0** ce primește un parametru **a** și un parametru **b**. Funcția va returna suma numerelor **naturale** din intervalul **[a, b)**.

În toate testele a va fi mai mic decât b.

Exemplu:
problema0(a=1, b=10) va returna **45**.

problema0(a=2, b=3) va returna **2**.

Test #	Input	Output	Expected	Passed	Score
0	1,10		45	<div>✗</div>	0

Problema 1 - Total: Puncte

Să se scrie o funcție cu numele **problema1** ce primește ca paramentru un string **my_str** în format **CamelCase** și întoarce stringul în format **lowercase_with_underscores**.

Un string este în format CamelCase atunci când primele litere din cuvinte sunt capitalizate.
Un string este în format lowercase_with_underscores atunci când toate cuvintele sunt doar cu litere mici și sunt separate prin underscore.

Stringul primit ca input conține doar litere.

"a".islower() == True
"A".islower() == False
"A".lower() == "a"

Exemplu:
problema1(my_str="CamelCase") va returna **"camel_case"**.

Test #	Input	Output	Expected	Passed	Score
0	"CamelCase"		camel_case	<div>✗</div>	0



Problema 2 - Total: Puncte

Să se scrie o funcție **problema2** ce primește ca parametru o listă de numere întregi **my_list** și care returnează o listă cu numerele din **my_list** ce apar de un număr impar de ori. **La final, lista returnată va conține doar elemente unice.**

Exemplu:
problema2(my_list=[1, 2, 3, 4, 1, 2, 2, 3, 4, 4, 4, 5]) va returna **[2, 5]**.
Nu contează ordinea din rezultat.

Test #	Input	Output	Expected	Passed	Score
0	[1, 2, 3, 1, 2, 3, 1, 5]		[1, 5]	<div>✖</div>	0

Problema 3 - Total: Puncte

Să se scrie o funcție **problema3** ce primește ca parametru o listă cu stringuri **my_list** și returnează un dicționar ce are ca și chei md5(string_din_lista) și ca valori string_din_lista pentru toate stringurile din **my_list** ce conțin substringuri de minim **5** cifre **pe poziții consecutive.**

Exemple:
problema3(my_list=["nota5"]) va returna {}

problema3(my_list=["ss12345", "54321cba", "a11111a", "2222a2", "a", "123456"]) va returna
{
'805065647bd5334acd87c84461354df2': 'ss12345',
'6c6dd77d90a81c183c96ea4f05fb736a': '54321cba',
'61ff9c07020edbdac7ae80482652830b': 'a11111a',
'e10adc3949ba59abbe56e057f20f883e': '123456'
}

Tineți cont de faptul că 'a'.encode() = b'a'.

Test #	Input	Output	Expected	Passed	Score
0	['ss12345', '54321cba', 'a11111a', '2222a2', 'a', '123456']		{'805065647bd5334acd87c84461354df2': 'ss12345', '6c6dd77d90a81c183c96ea4f05fb736a': '54321cba', '61ff9c07020edbdac7ae80482652830b': 'a11111a', 'e10adc3949ba59abbe56e057f20f883e': '123456'}	<div>✖</div>	0

Problema 4 - Total: Puncte

Să se scrie o funcție **problema4** ce primește ca parametru un url **my_url**.
La conținutul acestui **my_url** se găsește un fișier JSON de forma

```
{  
  "data" : "aici_va_fi_un_string",  
  "next" : "http://aici.va.fi.un.url"  
}
```

Faceți requesturi HTTP la adresa **my_url** și apoi la valorile din cheia "next" cât timp există cheia next.
Problema va returna o listă cu valorile câmpului **data** în ordinea accesării/parcugerii.

Exemplu:
La adresa **"http://un.link/a.txt"** se află următorul conținut:
{
 "data": "to_return"
}
problema4("http://un.link/a.txt") va returna **["to_return"]**.

La adresa **"http://un.link/b.txt"** se află următorul conținut:
{
 "data": "za_info",
 "next": "http://un.link/a.txt"

Choose File...

Token

Submit



```
}

```

problema4("http://un.link/b.txt") va returna ["za_info", "to_return"].

Test #	Input	Output	Expected	Passed	Score
0	"https://pastebin.com/raw/auJgyakH"		['to_return']	<div>✗</div>	0

Problema 5 - Total: Puncte

Să se scrie o funcție **problema5** ce primește ca parametri o listă **my_list** și o matrice (listă de liste cu dimensiune n x n) **my_matrix** cu numere întregi.

Se va parcurge matricea **my_matrix** în spirală, în sensul acelor de ceasornic, începând cu colțul din dreapta sus și se va întoarce o listă cu toate elementele din **my_list** obținute astfel: element = my_list[my_matrix[x][y]] unde x și y reprezintă coordonatele pentru pasul curent din parcugerea în spirală.

Atenție , lista de elemente este circular infinită.

Considerând l = [1,2,3,4,5,6,7] atunci l[0]= 1, l[-3] = 5, l[-10] = 5, l[4] = 5, l[11] = 5

Pentru un apel cu

my_list=["X",1,"3",22,55.7,("ai","vd")]

my_matrix=[[8,2,3],

[2,-3,4],

[3,4,5]]

parcugerea în spirală ar fi 3, 4, 5, 4, 3, 2, 8, 2, -3

și valoarea returnată va fi [22, 55.7, ("ai", "vd"), 55.7, 22, "3", "3", "3", 22]

Test #	Input	Output	Expected	Passed	Score
0	['X', 1, '3', 22, 55.7, ('ai', 'vd')], [[8, 2, 3], [2, -3, 4], [3, 4, 5]]		[22, 55.7, ('ai', 'vd'), 55.7, 22, '3', '3', '3', 22]	<div>✗</div>	0

Problema 6 - Total: Puncte

Să se scrie o funcție **problema6** ce poate primi ori paramentrul **my_file** ori parametrul **my_folder**.

În cazul în care primește parametrul **my_folder** funcția **va parcurge RECURSIV fișierele** din directorul primit ca și parametru, va identifica grupurile de fișiere ce au același hash pe conținut și va returna o listă cu **CALEA COMPLETĂ (ABSOLUTĂ)** a fișierelor din grupul cu cele mai multe fișiere.

Exemplu: dacă **my_folder** conține fișierele

a.txt (conținut "123"), b.txt (conținut "132"), c.txt (conținut "123")

funcția va returna ['/eval/my_folder/a.txt', '/eval/my_folder/c.txt']

În cazul în care primește **my_file**, acesta va fi calea către un fișier JSON de forma

```
{
"sha": "un checksum sha256 aici",
"path": "o cale catre un folder"
}
```

Funcția va returna o listă cu toate **NUMELE FIȘIERELOR** găsite la **path-ul** din json care au hash-ul sha256 **PE CONȚINUT** egal cu cel din fișierul JSON. Parcugerea fișierelor se va face în mod **NERECURSIV**.

Test #	Input	Output	Expected	Passed	Score
0	my_folder = "test_zip_p6/caz1"		['/eval/test_zip_p6/caz1/a.txt', '/eval/test_zip_p6/caz1/b.txt', '/eval/test_zip_p6/caz1/c.txt']	<div>✗</div>	0

Choose File... Token

Submit



1	my_file = "test zip p6/caz1/a.json"	['a.txt', 'b.txt', 'c.txt']	✖ 0
---	---	-----------------------------	------------

Problema 7 - Total: Puncte

Sa se scrie o functie cu numele **problema7** ce primeste ca si parametri un string **ip**, un numar **port** si un mesaj **text**. La adresa (**ip**, **port**) se gaseste un server TCP.

- Comunicare cu serverul se va realiza astfel:
- clientul trimite parametrul **text** serverului
 - clientul citeste mesajele de la server pana cand intalneste substring-ul **PY**

Pe baza mesajului primit, se va returna **suma(numar_procesat(int(numar)))** pentru fiecare numar natural extras din test.

Functia **numar_procesat(numar)** returneaza:

- numarul de biti de **1** de pe pozitii **prime** din numar, daca numarul primit ca si parametru este **prim**
- numarul de biti de **1** de pe pozitii **neprime** din numar, daca numarul primit ca si parametru **nu** este **prim**

Atentie: Indexarea reprezentarii binare se va face de la **dreapta** spre **stanga**. **Cel mai din dreapta bit are poziția 0**.

Exemplu: Fie urmatorul mesaj primit de la server **1AF2XD3zS-51qww23.5PY**. Lista de numere naturale este: [1, 2, 3, 51 ,23 ,5].

- Reprezentarea lor binare este:
- **1 = 01** => **1** bit setat
 - **2 = 10** => **0** biti setati
 - **3 = 11** => **0** biti setati
 - **51 = 110011** => **3** biti setati
 - **23 = 10111** => **1** bit setat
 - **5 = 101** => **1** bit setat

Rezultat = 1 + 0 + 0 + 3 + 1 + 1 = 6

Pentru apelul:
problema7("fiieval.tk", 6969, "06374a0d16f145a4b2a3d6ad0cbdd8d4") rezultatul va fi **6**.

Test #	Input	Output	Expected	Passed	Score
0	"fiieval.tk",6969,"06374a0d16f145a4b2a3d6ad0cbdd8d4"		6	✖ 0	