****

**MINISTERUL EDUCAȚIEI, CULTURII ȘI CERCETĂRII**

**AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică şi Microelectronică**

**Departamentul Informatică şi Ingineria Sistemelor**

**Calancea Catalin**

**MI-222**

**Raport**

**pentru lucrarea de laborator Nr.2**

***la cursul de “Programare Interactivă”***

Verificat:

lector universitar

Strună Vadim

**Chișinău – 20****23**

**CUPRINS**

[1.INTRODUCERE 3](#_Toc150768669)

[2.SARCINA 4](#_Toc150768671)

[3.SCRIPT 6](#_Toc150768672)

[4.CONCLUZIA 9](#_Toc150768673)

[4.WEBOGRAFIE 10](#_Toc150768674)

# 1.INTRODUCERE

# În cadrul acestui laborator, vom explora diverse aspecte ale programării în limbajul Python, concentrându-ne pe lucrul cu șiruri și liste. Laboratorul este structurat în mai multe sarcini, fiecare abordând un set distinct de probleme. Vom analiza manipularea șirurilor și a listelor, efectuând operații precum tăierea, extinderea și sortarea. De asemenea, vom investiga diferite operații pe liste, precum inserarea, eliminarea și calculul de proprietăți specifice.

# 2.SARCINA

**I. Rezolvați sarcinile, unde este necesar faceți explicații**

1. Ce rezultat vom obține după executarea următorului cod asupra șirului s=’ProgramareaInteractiva’?

a) s[:-1]

b) s[:1]

c) s[0:1]

d) s[:]

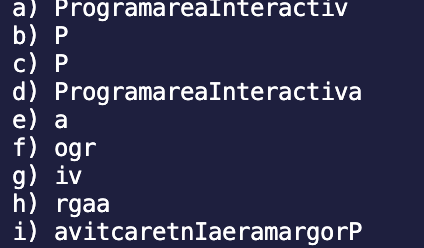
e) s[-1:]

f) s[2:5]

g) s[-3:-1]

h) s[1:9:2]

i) s[::-1]



2. Să presupunem că lista L conține elementele [’a’, ’b’, ’c’, ’d’, ’e’]. Să se evalueze valoarea expresiei:

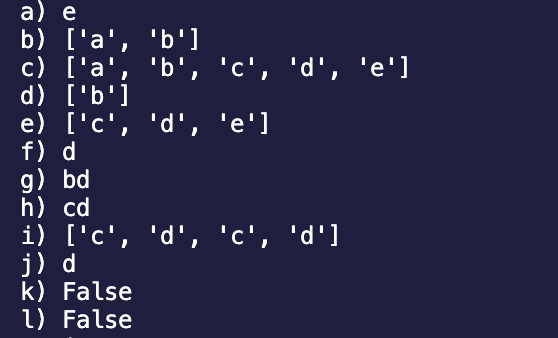
a) L [-1]  
b) L [:2]  
c) L [::]  
d) L [1:3:2]  
e) L [2:len(L)]

f) L [len(L)-2]

g)L[1]+L[3]  
h ) L[2]+ L [-2]  
i) 2\* L [2:4]  
j) L [int(int(‘3’\*2)/11))]

k) L[0:1]=[‘x’,’y’]

l) L[0:2]=[’Python’]



3. Ce rezultat vom obține după executarea următorului cod

a) b=[34,45,56,38]

b.insert(-1,12)

b) b=[34,45,56,38]

b.pop()  
c) b=[34,45,56,38]

b.remove(34)

d) b=[34,45,56,38]

b.append(‘doi’)

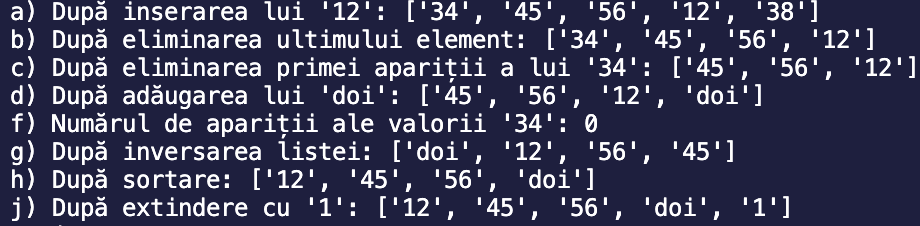
e) b.index(‘doi’)

f) b=[34,45,56,38]

b.count(34)

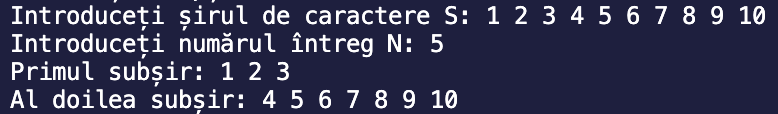
g) b.reverse()

h) b.sort()  
i) del(l[0:2])  
j) b.extend([1])



**II.Siruri:**

4. Sunt date de la tastatură un șir de caractere S și un număr întreg N. Să se obțină două subșiruri formate din intervalele [0,N] și (N,len()).



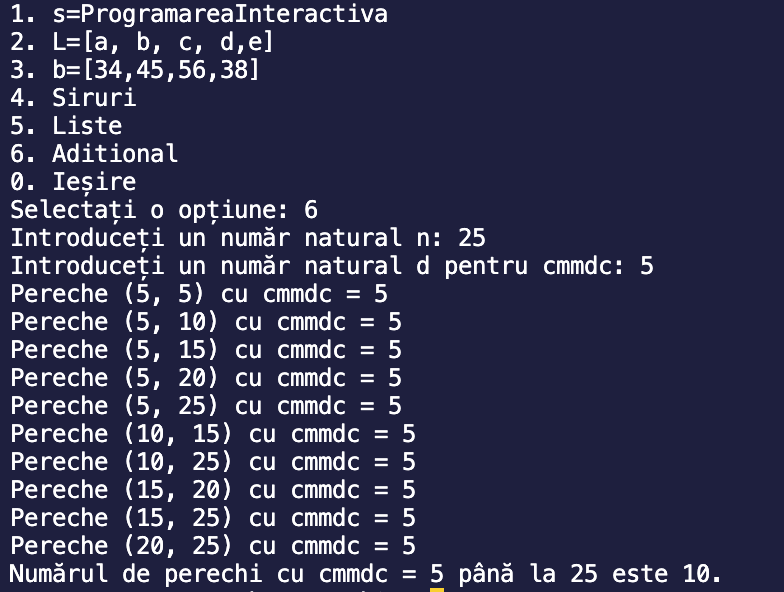
**III.Liste:**

4.Să se scrie un program, care calculează numărul de multipli de 4 prezenți în lista L=[4,34,56,74,68,98,52,62].



**III.Aditional:**

4. Să se afişeze numărul de perechi (şi perechile) de numere naturale care nu depăşesc un număr natural dat şi care au cel mai mare divizor comun un număr dat **d**.  
Exemplu: Pentru *n=20* există *6* perechi *(5, 5), (5, 10), (5, 15), (5, 20), (10, 15), (15, 20)* care au cel mai mare divizor comun egal cu *5*



# 3.SCRIPT

def main():

while True:

print("Meniu:")

print("1. s=ProgramareaInteractiva")

print("2. L=[a, b, c, d,e]")

print("3. b=[34,45,56,38]")

print("4. Siruri")

print("5. Liste")

print("6. Aditional")

print("0. Ieșire")

optiune = input("Selectați o opțiune: ")

if optiune == "1":

sarcina\_1()

elif optiune == "2":

sarcina\_2()

elif optiune == "3":

sarcina\_3()

elif optiune == "4":

sarcina\_4()

elif optiune == "5":

sarcina\_5()

elif optiune == "6":

sarcina\_6()

break

else:

print("Opțiune invalidă. Reîncercați.")

def sarcina\_1():

*# Sarcina 1*

s='ProgramareaInteractiva'

print("a)",s[:-1])

print("b)",s[:1])

print("c)",s[0:1])

print("d)",s[:])

print("e)",s[-1:])

print("f)",s[2:5])

print("g)",s[-3:-1])

print("h)",s[1:9:2])

print("i)",s[::-1])

def sarcina\_2():

*# Sarcina 2*

L = ['a','b','c','d','e']

print("a)",L[-1])

print("b)",L[:2])

print("c)",L[::])

print("d)",L[1:3:2])

print("e)",L[2:len(L)])

print("f)",L[len(L)-2])

print("g)",L[1]+L[3])

print("h)",L[2]+L[-2])

print("i)",2\*L[2:4])

print("j)",L [int(int('3'\*2)/11)])

print("k)",L [0:1]==['x','y'])

print("l)",L[0:2]==['Python'])

def sarcina\_3():

b = ['34', '45', '56', '38']

b.insert(-1, '12')

print("a) După inserarea lui '12':", b)

b.pop()

print("b) După eliminarea ultimului element:", b)

b.remove('34')

print("c) După eliminarea primei apariții a lui '34':", b)

b.append('doi')

print("d) După adăugarea lui 'doi':", b)

count\_34 = b.count('34')

print("f) Numărul de apariții ale valorii '34':", count\_34)

b.reverse()

print("g) După inversarea listei:", b)

b.sort()

print("h) După sortare:", b)

b.extend(['1'])

print("j) După extindere cu '1':", b)

def sarcina\_4():

*# Sarcina Siruri*

S = input("Introduceți șirul de caractere S: ")

N = int(input("Introduceți numărul întreg N: "))

if 0 <= N <= len(S):

subșir1 = S[:N]

subșir2 = S[N + 1:]

print("Primul subșir:", subșir1)

print("Al doilea subșir:", subșir2)

else:

print("Valoarea lui N este în afara intervalului corect pentru șirul dat.")

def sarcina\_5():

*# Sarcina Liste*

L = [4, 34, 56, 74, 68, 98, 52, 62]

numar\_multipli\_de\_4 = 0

for numar in L:

if numar % 4 == 0:

numar\_multipli\_de\_4 += 1

print("Numărul de multipli de 4 în lista L:", numar\_multipli\_de\_4)

def sarcina\_6():

*# Sarcina 6*

import math

def sarcina\_6():

numar\_perechi = 0

for i in range(1, n + 1):

for j in range(i, n + 1):

if math.gcd(i, j) == d:

numar\_perechi += 1

print(f"Pereche ({i}, {j}) cu cmmdc = {d}")

return numar\_perechi

n = int(input("Introduceți un număr natural n: "))

d = int(input("Introduceți un număr natural d pentru cmmdc: "))

rezultat = sarcina\_6()

print(f"Numărul de perechi cu cmmdc = {d} până la {n} este {rezultat}.")

if \_\_name\_\_ == "\_\_main\_\_": *#script este executat ca program principal*

main()

# 4.CONCLUZIA

În urma parcurgerii acestui laborator, am acumulat cunoștințe extinse despre manipularea șirurilor și listelor în Python. Am învățat să efectuăm operații precum tăierea șirurilor, accesarea elementelor listelor, precum și aplicarea de operații specifice, cum ar fi sortarea și extinderea. De asemenea, am rezolvat probleme practice, cum ar fi identificarea și calculul de multipli sau determinarea perechilor cu cel mai mare divizor comun. Aceste abordări ne permit să dezvoltăm competențe esențiale pentru manipularea eficientă a datelor în Python.

# 4.WEBOGRAFIE

1. Python Documentation. Disponibil online la: https://docs.python.org/

2. Stack Overflow. Comunitatea programatorilor. Disponibil online la: https://stackoverflow.com/

3. W3Schools. Tutoriale Python. Disponibil online la: https://www.w3schools.com/python/

4. GeeksforGeeks. Resurse Python. Disponibil online la: https://www.geeksforgeeks.org/python-programming-language/