

MINISTERUL EDUCAȚIEI, CULTURII și CERCETĂRII al Rep. MOLDOVA
UNIVERSITATEA TEHNICĂ a MOLDOVEI
FACULTATEA CALCULATOARE, INFORMATICĂ și MICROELECTRONICĂ
Departamentul „Inginerie Software si Automatica”

LUCRARE INDIVIDUALĂ GRAFICA PE CALCULATOR



Student(ă): COJOCARI Dragos_____

gr.TI-214, FCIM

Conducător:

Conf. univ. MALCOCI Iulian _____

CHIȘINĂU 2022

CUPRINS

CUPRINS	1
Tema 4. Operatori logici si de comparatie. Expresii. Structuri repetitive.....	2
Tema 5. Functiile in Pyhton	18
Exemple Turtle.....	27
Calculatorul.....	34

					GC Nr. 21-186 – Cojocari Dragoș		
Mo	Coal	Nr.	Semn.	Data	Lucrare independentă la disciplina: Grafica pe calculator		
Elaborat	Cojocari						
Verificat	Malcoci						
					Litera	Coala	Coli
						1	34
					UTM FCIM Gr. TI-214		

Tema 4. Operatori logici si de comparatie. Expresii. Structuri repetitive

Ex4_1

Ce răspuns vom obține după rularea următorului cod?

```
print ("Python" == "PYTHON")
print ("Python" != "PYTHON")
print ("PYTHON" != "PYTHON")
print (15 <= 14)
print (15 <= 15)
```

```
False
True
False
False
True
```

Ex4_2

Ce răspuns vom obține după rularea următorului cod?

```
print('Welcome' != 'WELCOME' and (7>5) and (4==4))
print('Welcome' != 'WELCOME' or (7>5) or (4==4))
print('Welcome' == 'WELCOME' and (7<5) and (4==4))
print('Welcome' == 'WELCOME' or (7<5) or (4==4))
print('Welcome' == 'WELCOME' or (7<5) and (4==4))
print('Welcome' == 'WELCOME' and (7<5) or (4==4))
```

```
True
True
False
True
False
True
```

Ex4_3

Ce răspuns vom obține după rularea următorului cod?

```
x = 15
y = 7

print(x == y)
print(x != y)
print(x > y)
print(x >= y)
print(x == y and x != y)
print(x == y or x != y)
```

```
False
True
True
True
False
True
```

Ex4_4

Să presupunem că avem următorul dicționar `my_dict = {'key 1': 1, 'key 2': 7, 'key 3': 9}`, trebuie să creăm un cod care la răspuns să ne dea o listă cu valorile cheilor din dicționar care sunt mai mari decât 5.

```
my_dict = {'key 1': 1, 'key 2': 7, 'key 3': 9}
new_list = []
x = list(my_dict.values())
for i in x:
    if i >= 5:
        new_list.append(i)
print(new_list)
```

```
[7, 9]
```

Ex4_5

În acest exemplu trebuie să generați un număr aliator `a` între 1 și 100 (sugestie: importam modulul `random` și folosim `.randint(1,100)`), să cereți de la utilizator să introducă un număr `b` între 1 și 100, după care să comparăm aceste două numere.

```
import random
a = random.randint(1,100)
b = int(input( " Alege un numar intre 1 si 100 : "))

if (a > b) :
    print(F" Numarul a > b : {a} > {b} ")
elif (a<b) :
    print(F" Numarul a < b : {a} < {b} ")
else :
    print(F" Numarul a = b : {a} = {b} ")
```

```
Alege un numar intre 1 si 100 : 5
Numarul a > b : 96 > 5
```

					GC Nr. 21-186 – Cojocari Dragoș	Coal
Mo	Coal	N.	Semnat	Data		3

Ex4_6

În acest exemplu trebuie să scrieți un cod în care să cereți de la utilizator să introducă orice număr dorește, iar la răspuns să obținem dacă numărul ales este pozitiv, negativ sau egal cu zero.

```
num = float(input("Introduceti numarul : "))
if num >= 0:
    if num == 0: print("Zero")
    else: print("Numarul este pozitiv")
else: print("Numarul este negativ")
```

```
Introduceti numarul : 12
Numarul este pozitiv
```

Ex4_7

În acest exemplu trebuie să scrieți un cod în care să cereți de la utilizator să introducă un număr între 1 și 10000, după care să verificăm dacă acest număr este multiplul lui 5 dar nu și multiplul lui 3.

```
4 b = int(input( " alege un numar intre 1 si 100 : "))
3 if (b%5==0) and (b%3!=0) : print(F"Numarul este multiplul lui 5 si 3 {b} ")
2 else : print ("Condițiile nu se respecta ")
```

```
alege un numar intre 1 si 100 : 125
Numarul este multiplul lui 5 si 3 125
```

Ex4_8

În acest exemplu trebuie să scrieți un cod în care să cereți de la utilizator să introducă un număr pozitiv, iar la răspuns să obținem din câte cifre este format numărul ales de utilizator. De exemplu dacă utilizatorul a ales numărul 77 la răspuns trebuie să obținem: Nr ales de utilizator are 2 cifre.

```
nr = int(input("Introduceti numarul : "))
x = len(str(nr))
print(F"Nr ales de utilizator are {x} cifre")
```

```
Introduceti numarul : 123
Nr ales de utilizator are 3 cifre
```

Ex4_9

În acest exemplu trebuie să scrieți un cod în care să cereți de la utilizator să introducă un număr oarecare, iar la răspuns să obținem răspunsul dacă acest număr este par sau impar.

```
nr = int(input("Introduceti numarul : "))

if nr%2 == 0 :
    print ( "Numarul este par")
else:
    print(" Numarul este impar")
```

Introduceti numarul : 5
Numarul este impar

Ex4_10

Pentru dicționarul `my_dict = {'Ana': 11, 'Mariana': 40, 'Iulian': 41}` scrieți un cod care la răspuns să ne dea o listă care să conțină doar valorile cheilor din dicționar care sunt mai mari de 25.

```
my_dict = {'Ana': 11, 'Mariana': 40, 'Iulian': 41}
new_list = []
x = list(my_dict.values())
for i in x:
    if i >= 25:
        new_list.append(i)
print(new_list)
```

[40, 41]

Ex4_11

Care va fi răspunsul după rularea următoarelor coduri:

a)

```
lista_mea = ['Ion', 'Ana', 'Maria']
for x in lista_mea:
    print(x)
    print('Salut')
```

b)

```
lista_mea = ['Ion', 'Ana', 'Maria']
for x in lista_mea[0:2]:
    print(x)
    print('Salut')
```

c)

```
lista_mea = ['Ion', 'Ana', 'Maria']
for x in lista_mea:
    print(x)
    print('Salut')
```

					GC Nr. 21-186 – Cojocari Dragoș	Coal
Mo	Coal	N.	Semnat	Data		5

```

# a)

lista_mea = ['Ion', 'Ana', 'Maria']
for x in lista_mea:
    print(x)
    print('Salut')

print()
# b)

lista_mea = ['Ion', 'Ana', 'Maria']
for x in lista_mea[:2]:
    print(x)
    print('Salut')

print()
# c)

lista_mea = ['Ion', 'Ana', 'Maria']
for x in lista_mea:
    print(x)
print('Salut')

```

```

Ion
Salut
Ana
Salut
Maria
Salut

Ion
Salut
Ana
Salut

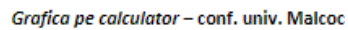
Ion
Ana
Maria
Salut

```

Care va fi răspunsul după rularea următoarelor coduri:

```
lista_mea = ['Ion', 'Ana', 'Maria']
for x in lista_mea:
    print(x)
```

2



```
# a)

lista_mea = ['Ion', 'Ana', 'Maria']
for x in 'Ana':
    print(x)

print()

# b)

lista_mea = ['Ion', 'Ana', 'Maria']
for x in 'Ana':
    print(x, end = '')
```

Ana

Care va fi răspunsul după rularea următorului cod:

```
i=0
while i < len(lista_mea):
    telefoane = lista_mea[i]
    i += 1

    if telefoane == 'google':
        continue

    print(telefoane)
```

					GC Nr. 21-186 – Cojocari Dragoș	Coal
						7
Mo	Coal	N.	Semnăt	Data		


```

lista_mea = ['nokia', 'samsung', 'google', 'iphone']
i = 0
while i < len(lista_mea) :
    telefoane = lista_mea[i]
    i +=1
    if telefoane == 'google':
        continue
    print(telefoane)

```

```

nokia
samsung
iphone

```

Ex4_14

Scrieți un cod folosind bucla for și al doilea cod folosind bucla while astfel încât la răspuns să obținem:

```

1
2
3
4
5
6
7

```

```

# a)
for i in range(1,8):
    print(i)

print()
# b)
i = 1
while i < 8:
    print(i)
    i +=1

```

```

1
2
3
4
5
6
7
1
2
3
4
5
6
7

```

Ex4_15

Scrieți un cod folosind bucla for astfel încât la răspuns să obține 3 coloane, în prima coloană să fie numerele de la 1 la 10, în a doua coloană să fie pătratul acestor numere, iar a treia coloană să fie cubul acestor numere ca în exemplu:

a)

```
1 1 1
2 4 8
3 9 27
4 16 64
5 25 125
6 36 216
7 49 343
8 64 512
9 81 729
10 100 1000
```

b)

```
1 1 1
2 4 8
3 9 27
4 16 64
5 25 125
6 36 216
7 49 343
8 64 512
9 81 729
10 100 1000
```

```
for i in range(1,11):
    print('{} {} {}'.format(i , i**2 , i**3))

print()
# b)

for i in range(1,11):
    print('{0:2d} {1:3d} {2:4d}'.format(i , i**2 , i**3))
```

```
1 1 1
2 4 8
3 9 27
4 16 64
5 25 125
6 36 216
7 49 343
8 64 512
9 81 729
10 100 1000
```

```
1 1 1
2 4 8
3 9 27
4 16 64
5 25 125
6 36 216
7 49 343
8 64 512
9 81 729
10 100 1000
```

Mo	Coal	N.	Semnat	Data

GC Nr. 21-186 – Cojocari Dragoș

Coal

9

Ex4_16

Scrieți un cod care trebuie să simuleze aruncarea zarurilor cu ajutorul buclei while și modulului random. Răspunsul trebuie să arate în felul următor:

```
***** ARUNCĂ ZARURILE *****
Zarurile sunt aruncate asteapta ...
Zarul 1 indică: 4
Zarul 2 indică: 2
Apasa tasta y pentru a arunca din nou zarurile: y
Zarurile sunt aruncate asteapta ...
Zarul 1 indică: 1
Zarul 2 indică: 4
Apasa tasta y pentru a arunca din nou zarurile:
```

```
print('*'*5, 'Arunca zarurile' , '*'*5 )
import random
import time
zar = 'y'
while (zar == 'y'):
    print(" Aruncam zarurile ")
    print("Zar1 indica : ", random.randint(1,6))
    print("Zar2 indica : ", random.randint(1,6))
    zar = input("Apasa tasta y pentru a arunca zarurile din nou : ")
    time.sleep(3)
```

```
***** Arunca zarurile *****
Aruncam zarurile
Zar1 indica : 5
Zar2 indica : 3
Apasa tasta y pentru a arunca zarurile din nou : y
Aruncam zarurile
Zar1 indica : 4
Zar2 indica : 6
Apasa tasta y pentru a arunca zarurile din nou :
```

Ex4_17

Trebuie să scriem un cod care să ceară de la utilizator să introducă un număr de la 1 până la 9, după care folosind bucla for să obținem tabla înmulțirii cu acel număr ales de utilizator. Răspunsul trebuie să arate în felul următor:

```
Alege un numar cuprins între 1 si 9: 7
7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
7 x 4 = 28
7 x 5 = 35
7 x 6 = 42
7 x 7 = 49
7 x 8 = 56
7 x 9 = 63
7 x 10 = 70
```

```
num = int(input("Alege un numar cuprins între 1 si 9 :"))
for i in range(1 , 10) :
    x = num* i
    print(F"{num} x {i} = {x}")
```

					GC Nr. 21-186 – Cojocari Dragoș	Coal
Mo	Coal	N.	Semnat	Data		10

```
Alege un numar cuprins intre 1 si 9 :7
7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
7 x 4 = 28
7 x 5 = 35
7 x 6 = 42
7 x 7 = 49
7 x 8 = 56
7 x 9 = 63
```

Ex4_18

Trebuie să scriem un cad care să ceară de la utilizator o valoare minimă și una maximă, iar după rulare la răspuns să apară doar numerele pare dintre aceste două valori alese de utilizator. Răspunsul trebuie să arate în felul următor:

```
Introdu limita de jos: 45
Introdu limita de sus: 63
46
48
50
52
54
56
58
60
62
```

```
a = int ( input( " introdu limita de jos :"))
b = int ( input( " introdu limita de sus :"))
my_list = list(range(a,b))
for i in my_list:
    if i % 2 == 0:
        print(i)
```

```
intodu limita de jos :46
intodu limita de sus :63
46
48
50
52
54
56
58
60
62
```

					GC Nr. 21-186 – Cojocari Dragoș	Coal
Mo	Coal	N.	Semnat	Data		11

Ex4_19

Ce răspuns vom obține dacă vom rula următorul cod:

```
for i in 'Dacia':  
    if i == 'a':  
        continue  
    print(i)
```

```
D  
c  
i
```

Ex4_20

Trebuie să scriem un cod în care să cerem de la utilizator să introducă câteva valori separate prin spațiu pentru temperatura în grade C sub formă de listă, după care această listă să o transformăm într-o listă nouă care să ne arate aceste valori în grade F cu ajutorul buclei for. Răspunsul trebuie să arate în felul următor:

```
Introdu 4 valori pentru temp in grade C separate prin spatiu  
0 20 30 45  
Temperatura in grade C si grade F  
[0, 20, 30, 45] [32.0, 68.0, 86.0, 113.0]
```

```
print("Introdu 4 valori pentru temp in grade C separate prin spatiu ")  
Temp_C = [int(x) for x in input().split()]  
Temp_F = [(9/5 *y +32) for y in Temp_C]  
print("Temperatura in grade C si grade F\n",Temp_C,Temp_F)
```

```
Introdu 4 valori pentru temp in grade C separate prin spatiu  
0 20 30 45  
Temperatura in grade C si grade F  
[0, 20, 30, 45] [32.0, 68.0, 86.0, 113.0]
```

Ex4_21

Trebuie să scriem un cod în care utilizatorul să specifice lungimea parolei pe care calculatorul trebuie s-o genereze în mod aleatoriu cu ajutorul modului random și folosirea buclei for, iar caracterele să conțină litere, numere și caractere speciale din modulul string.

```
import string  
print('Numere:', string.digits)  
print('Litere:', string.ascii_letters)  
print('Simboluri:', string.punctuation)
```

După rulare vedem ce conține fiecare instrucțiune din modulul string:

```
Numere: 0123456789  
Litere: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ  
Simboluri: !"#%&'()*+,-./:;<=>?@[\]^_`{|}~
```

După ce terminăm de scris codul care va genera o parolă în mod aleatoriu vom obține un răspuns care arată în felul următor:

```
Introdu lungimea parolei: 5  
sdAsI
```

					GC Nr. 21-186 – Cojocari Dragoș	Coal
Mo	Coal	N.	Semnat	Data		12

```
import string
import random
parola = int(input("Introdu lungimea parolei : "))
caractere = string.digits + string.ascii_letters + string.punctuation
PAROLA = " "
for i in range(0,parola):
    PAROLA = PAROLA + random.choice(caractere)
print(PAROLA)
```

```
Introdu lungimea parolei : 5
bZADn
```

Ex4_22

Trebuie să scriem un cod care să îl întrebe pe utilizator cum se numește și câți ani are, după care utilizând comenzile `.split()`, `.sort()` și bucla `for` să sortăm cuvintele din propoziția introdusă de utilizator. După rularea codului răspunsul trebuie să arate în felul următor:

```
Cum te numesti si ce vârstă ai?: Mă numesc Iulian Malcoci si am 41 de ani.
41
Iulian
Malcoci
Mă
am
ani.
de
numesc
si
```

```
a = input("Cum te numesti si ce varsta ai :")
b = a.split()
b.sort()
for i in b:
    print(i)
```

```
Cum te numesti si ce varsta ai :Ma numesc Iulian Malcoci si am 41 de ani
41
Iulian
Ma
Malcoci
am
ani
de
numesc
si
```

Ex4_23

Având două liste `culori = ['Alb', 'Negru', 'Oranj']` și `auto = ['Duster', 'Logan', 'Dokker']` trebuie să scriem două coduri în care să folosim bucle imbricate pentru a obține următoarele răspunsuri:

a)

					GC Nr. 21-186 – Cojocari Dragoș	Coal
Mo	Coal	N.	Semnat	Data		13

```
Duster Alb
Logan Alb
Dokker Alb
Duster Negru
Logan Negru
Dokker Negru
Duster Oranj
Logan Oranj
Dokker Oranj
```

b)

```
[('Duster', 'Alb'), ('Logan', 'Alb'), ('Dokker', 'Alb'), ('Duster', 'Negru'),
('Logan', 'Negru'), ('Dokker', 'Negru'), ('Duster', 'Oranj'), ('Logan', 'Oranj'), ('Dokker', 'Oranj')]
```

```
culori = ['Alb', 'Negru', 'Oranj']
auto = ['Duster', 'Logan', 'Dokker']
for i in culori:
    for q in auto :
        print(q,i)
print()
lista = [(q,i)for i in culori for q in auto ]
print(lista)
```

```
Duster Alb
Logan Alb
Dokker Alb
Duster Negru
Logan Negru
Dokker Negru
Duster Oranj
Logan Oranj
Dokker Oranj
```

```
[('Duster', 'Alb'), ('Logan', 'Alb'), ('Dokker', 'Alb'), ('Duster', 'Negru'),
('Logan', 'Negru'), ('Dokker', 'Negru'), ('Duster', 'Oranj'), ('Logan', 'Oranj'), ('Dokker', 'Oranj')]
```

Ex4_24

Trebuie să scriem un cod care să ceară de la utilizator să introducă o propoziție, după care cu ajutorul buclei for să obținem la răspuns nr. de litere mici și nr. de litere mari. Răspunsul trebuie să arate în felul următor:

```
Introdu o propozitie: Republica Moldova
Numărul de litere mari este: 2
Numărul de litere mici este: 14
```

					GC Nr. 21-186 – Cojocari Dragoș	Coal
Mo	Coal	N.	Semnat	Data		14

```

sir = input (" Introdu o propozitie: ")
litere_mari = 0
litere_mici = 0
for i in sir :
    if i.isupper():
        litere_mari = litere_mari + 1
    elif i.islower():
        litere_mici +=1
    else:
        pass
print(" Numarul de litere mari este : ", litere_mari)
print(" Numarul de litere mici este : ", litere_mici)

```

```

Introdu o propozitie: Republica Moldova
Numarul de litere mari este : 2
Numarul de litere mici este : 14

```

Ex4_25

Trebuie să scriem un cod în care să cerem de la utilizator să introducă limita de jos și limita de sus și limita de jos între 100 și 999. După care la răspuns să obținem o listă care să cuprindă o listă doar din numerele care au toate cele trei cifre numere pare. Răspunsul trebuie să arate în felul următor:

```

Introdu limita de jos începând cu 100: 123
Introdu limita de sus care nu depășeste 999: 217
['200', '202', '204', '206', '208']

```

```

x_min = int(input("Introdu limita de jos incepand cu 100 : "))
x_max = int(input("Introdu limita de care nu depaseste 999 : "))
numere_pare = []
for i in range (x_min,x_max):
    s =str(i)
    if (int(s[0])%2 == 0 and (int(s[1])%2 == 0) and (int(s[2])%2==0)) :
        numere_pare.append(s)
print(numere_pare)

```

```

Introdu limita de jos incepand cu 100 : 123
Introdu limita de care nu depaseste 999 : 217
['200', '202', '204', '206', '208']

```

					GC Nr. 21-186 – Cojocari Dragoș	Coal
Mo	Coal	N.	Semnat	Data		15

Ex4_26

În acest exercițiu vom folosi modulul `time` și comanda `.sleep()` pentru ca de exemplu răspunsul dorit de noi să apară pe ecran să apară la fiecare 5 secunde. Vom întreba care este filmul preferat al utilizatorului. După rulare la fiecare 5 secunde va apărea mesajul: Acest film este SUPER!!! După rulare vom obține:

```
Care este filmul tau preferat?: Tunul de lemn
```

```
Tunul de lemn este un film SUPER!!!
```

```
Tunul de lemn este un film SUPER!!!
```

```
Tunul de lemn este un film SUPER!!!
```

```
Tunul de lemn este un film SUPER!!!
```

```
Tunul de lemn este un film SUPER!!!
```

```
Traceback (most recent call last):
```

```
File "C:\Users\Admin\Desktop\exemplu.py", line 7, in <module>
```

```
time.sleep(5)
```

```
KeyboardInterrupt
```

Codul va rula la nesfârșit și la fiecare 5 secunde va apărea un nou mesaj Tunul de lemn este un film SUPER!!! Dacă dorim să întrerupem rularea apăsăm tastele Ctrl + C.

```
import time
film = input("Care este filmul tau preferat ? : ")
while True:
    print(film, 'este un film SUPER !!!')
    time.sleep(5)
```

```
Care este filmul tau preferat ? : Tunul de lemn
```

```
Tunul de lemn este un film SUPER !!!
```

```
Tunul de lemn este un film SUPER !!!
```

```
Tunul de lemn este un film SUPER !!!
```

```
Traceback (most recent call last):
```

```
File "C:\Users\xccelerator\Desktop\lab4-5\Lab4.py", line 272, in <module>
```

```
time.sleep(5)
```

```
KeyboardInterrupt
```

```
^C
```

Ex4_27

În acest exercițiu îi vom cere utilizatorului să introducă o parolă cu condițiile ca această parolă să fie formată din 10 caractere, să conțină cel puțin o literă mare și cel puțin o cifră. Aici vom folosi cam tot ce am învățat până acum și bucla `while` și bucla `for` și operatorul logic `and` și instrucțiunile condiționate `if` și `else`. După rulare răspunsul poate arăta în felul următor:

```
Introdu parola: qwertya555
```

```
Parola nu are 10 caractere, o cifra sau majusculă!!!
```

```
Introdu parola: Qwertya555
```

```
Parola nouă a fost salvată cu succes
```

```
7 while True:
6     parola = input("Introdu parola : ")
5     if len(parola)>= 10 and any(i.isdigit() for i in parola) and any(i.is
upper() for i in parola):
4         print("Parola noua a fost salvata cu succes")
3         break
2     else:
1         print("Parola nu are 10 caractere , o cifra sau majuscula !!!")
```

					GC Nr. 21-186 – Cojocari Dragoș	Coal
Mo	Coal	N.	Semnat	Data		16

```
Introdu parola : qwertya555
Parola nu are 10 caractere , o cifra sau majuscula !!!
Introdu parola : Qwertya555
Parola noua a fost salvata cu succes
```

					GC Nr. 21-186 – Cojocari Dragoș	Coal
						17
Mo	Coal	N.	Semnat	Data		

Tema 5. Functiile in Pyhton

Ex5_1

Trebuie să creăm o funcție care să redea pătratul oricărui număr, iar numărul respectiv să fie introdus de către utilizator cu funcția `input()`. După rularea codului răspunsul trebuie să arate în felul următor:

```
Introdu un număr: 4  
Pătratul numărului 4 este 16
```

```
a = int(input("Introduceti numarul dorit: "))  
b = lambda a : a**2  
print(f"Patratal numarului {a} este {b(a)}")
```

```
Introduceti numarul dorit: 4  
Patratal numarului 4 este 16
```

Ex5_2

Trebuie să creăm o funcție care să redea suma și produsul a două numere. Numerele trebuie să fie introduse de utilizator. După rularea codului răspunsul trebuie să arate în felul următor:

```
Introdu primul număr: a=5  
Introdu al doilea număr: b=3  
Suma a+b=8, Produsul a*b=15  
Suma 5+3=8, Produsul 5*3=15
```

```
9 a = int(input("Introdu primul numar: a = "))  
8 b = int(input("Introdu al 2-lea numar: b = "))  
7  
6 def f(a,b):  
5     print(f"Suma a+b={a+b}, Produsul a*b={a*b}")  
4     print(f"Suma {a}+{b}={a+b}, Produsul {a}*{b}={a*b}")  
3 )  
3 f(a,b)
```

```
Introdu primul numar: a = 5  
Introdu al 2-lea numar: b = 3  
Suma a+b=8, Produsul a*b=15  
Suma 5+3=8, Produsul 5*3=15
```

					GC Nr. 21-186 – Cojocari Dragoș	Coal
Mo	Coal	N.	Semnat	Data		18

Ex5_3

Trebuie să realizăm un cod care să redea suma și produsul a două numere folosindu-ne de expresia lambda. Numerele trebuie să fie introduse de utilizator. După rularea codului răspunsul trebuie să arate în felul următor:

```
Introdu primul număr: a=5
Introdu al doilea număr: b=3
Suma a+b=8, Produsul a*b=15
Suma 5+3=8, Produsul 5*3=15
```

```
7 a = int(input("Introdu primul numar: a = "))
6 b = int(input("Introdu al 2-lea numar: b = "))
5 c = lambda a,b:a+b
4 d = lambda a,b:a*b
3 print(f"Suma a+b={c(a,b)}, Produsul a*b={d(a,b)}")
2 print(f"Suma {a}+{b}={c(a,b)}, Produsul {a}*{b}={d(a,b)}")
```

```
Introdu primul numar: a = 5
Introdu al 2-lea numar: b = 3
Suma a+b=8, Produsul a*b=15
Suma 5+3=8, Produsul 5*3=15
```

Ex5_4

Trebuie să realizăm o funcție care să redea după rulare factorialul numărului 4, iar numărul 4 să fie introdus de utilizator ($4 \times 3 \times 2 \times 1 = 24$). După rularea codului răspunsul trebuie să arate în felul următor:

```
Doriti sa determinati factorialul numărului: 4
Factorialul lui 4 este 24
```

```
10 a = int(input("Doriti sa determinati factorialul numaru
    lui:"))
9 b = 1
8 def factorial(a):
7     b = 1
6     for i in range(1,a+1):
5         b = b * i
4     return b
3 c = factorial(a)
2 print(f"Factorialul lui {a} este {c}")
```

```
Doriti sa determinati factorialul numărului:4
Factorialul lui 4 este 24
```

Ex5_5

Presupunând că avem creată o listă cu funcția `lista_mea = range(-7,8)`, trebuie să creăm un cod care după rulare să redea o nouă listă în care toate elementele din lista inițială să fie ridicate la cub.

`[-343, -216, -125, -64, -27, -8, -1, 0, 1, 8, 27, 64, 125, 216, 343]`

```
lista_mea = list(range(-7,8))
lista_noua = []
for i in lista_mea:
    i = i**3
    lista_noua.append(i)
print(lista_noua)
```

`[-343, -216, -125, -64, -27, -8, -1, 0, 1, 8, 27, 64, 125, 216, 343]`

Ex5_6

Presupunând că avem deja scris următorul cod:

```
def patrat(x):
    return x*x
```

`numere = range(-5,6)`

folosind expresia lambda trebuie să continuăm acest cod astfel ca la răspuns să obținem o listă care să conțină toate elementele din variabila `numere` ridicate la pătrat. După rulare trebuie să obținem următorul răspuns:

Pătratul listei

`[-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5]` este
`[25, 16, 9, 4, 1, 0, 1, 4, 9, 16, 25]`

```
def patrat(x):
    return x*x
numere = list(range(-5,6))
patr = []
for i in numere:
    a = lambda i : patrat(i)
    i = a(i)
    patr.append(i)
print(patr[:])
```

`[25, 16, 9, 4, 1, 0, 1, 4, 9, 16, 25]`

					GC Nr. 21-186 – Cojocari Dragoș	Coal
Mo	Coal	N.	Semnat	Data		20

Ex5_7

Trebuie să creăm 3 funcții care să dea diametrul, lungimea și aria unui cerc la răspuns. Raza cercului trebuie să fie introdusă de utilizator. După rulare trebuie să obținem următorul răspuns:

```
Care este raza cercului?: 4
Diametrul cercului = 8
Lungimea cercului = 25.132741228718345
Aria cercului = 50.26548245743669
```

```
a = int(input("Introduceti raza cercului:"))
b = lambda b : b*2
print(f"Diametrul cercului cu raza {a} este {b(a)}")
c = lambda c : 2 * 3.14 * a
print(f"Lungimea cercului cu raza {a} este {c(a)}")
d = lambda d : 3.14*a*a
print(f"Aria cercului cu raza {a} este {d(a)}")
```

```
Introduceti raza cercului:4
Diametrul cercului cu raza 4 este 8
Lungimea cercului cu raza 4 este 25.12
Aria cercului cu raza 4 este 50.24
```

Ex5_8

Trebuie să creăm un cod în care să cerem de la utilizat limita minimă (număr negativ) și limita maximă, din șirul de numere obținut să obținem la răspuns o listă care să cuprindă doar numerele pozitive și pare (trebuie să folosim funcțiile range(), filter() și expresia lambda. După rulare trebuie să obținem următorul răspuns:

```
Introdu limita minimă (nr. negativ): -7
Introdu limita maximă (nr. pozitiv): 27
[2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26]
```

```
a = int(input("Introdu limita minima(nr.negativ):"))
b = int(input("Introdu limita maxima(nr.negativ):"))
lista_initiala = list(range(a,b))
lista_para = filter(lambda d:d%2==0 and d > 0, lista_initiala)
print(list(lista_para))
```

```
Introdu limita minima(nr.negativ):-7
Introdu limita maxima(nr.negativ):27
[2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26]
```

Ex5_9

Un exemplu asemănător cu cel precedent. Dacă avem lista numere = [-1, 2, 45, 20, -23, 17, -3, 9, -5, 6, 32, -14, 14, 17, 12, -20, 11, 8] trebuie să scriem un cod folosind funcția filter() și expresia lambda astfel încât la răspuns să obținem două liste noi, prima listă să conțină doar numerele negative și cele impare, iar a doua listă să conțină numerele mai mari decât -20 și mai mici decât 20. După rulare trebuie să obținem următorul răspuns:

Nr negative / impare: [-1, -23, -3, -5]

Nr între -20 și 20: [-1, 2, 17, -3, 9, -5, 6, -14, 14, 17, 12, 11, 8]

```
lista_initiala=[-1,2,45,20,-23,17,-3,9,-5,6,32,-14,14,17,12,-20,11,8]
lista_negativa_impura = filter(lambda a:a%2!=0 and a<0, lista_initiala)
print(f"Nr negative/impura: {list(lista_negativa_impura)}")
lista_20_20= filter(lambda b:b>-20 and b<20, lista_initiala)
print(f"Nr între -20 și 20: {list(lista_20_20)}")
```

Nr negative/impura: [-1, -23, -3, -5]

Nr între -20 și 20: [-1, 2, 17, -3, 9, -5, 6, -14, 14, 17, 12, 11, 8]

Ex5_10

Să presupunem că avem două matrice $x = \begin{bmatrix} 3 & 9 \\ 8 & 5 \end{bmatrix}$ și $y = \begin{bmatrix} 2 & 3 \\ 1 & 7 \end{bmatrix}$. Trebuie să creăm un cod în care să definim funcții pentru adunarea (np.add(x,y)), scăderea (np.subtract(x,y)), înmulțirea (np.multiply(x,y)), împărțirea (np.divide(x,y)), înmulțirea scalară (np.dot(x,y)) a matricelor x și y și funcția pentru determinarea transpusei matricei x (x.T) și inversei matricei y (np.linalg.inv(y)). După rulare trebuie să obținem următorul răspuns:

```
Suma matricelor x+y=
[[ 5 12]
 [ 9 12]]
Diferenta matricelor x-y=
[[ 1  6]
 [ 7 -2]]
Produsul matricelor x*y=
[[ 6 27]
 [ 8 35]]
Împărțirea matricelor x/y=
[[1.5      3.      ]
 [8.      0.71428571]]
Transpusa matricei x=
[[3 8]
 [9 5]]
Inversa matricei y=
[[ 0.63636364 -0.27272727]
 [-0.09090909  0.18181818]]
```

```
import numpy as np
a = np.array(((3,9),(8,5)))
b = np.array(((2,3),(1,7)))
c = a+b
print("Suma matricelor x+y=")
print(c)
c = a-b
print("Diferenta matricelor x-y=")
print(c)
c = a*b
print("Produsul matricelor x*y=")
print(c)
c = a/b
print("Impartirea matricelor x/y=")
print(c)
print("Transpusa matricei x = ")
print(a.transpose())
g=np.linalg.inv(b)
print("Inversa matricei y=")
print(g)
```

					GC Nr. 21-186 – Cojocari Dragoș	Coal
Mo	Coal	N.	Semnat	Data		22


```

Suma matricelor x+y=
[[ 5 12]
 [ 9 12]]
Diferenta matricelor x-y=
[[ 1  6]
 [ 7 -2]]
Produsul matricelor x*y=
[[ 6 27]
 [ 8 35]]
Impartirea matricelor x/y=
[[1.5      3.      ]
 [8.      0.71428571]]
Transpusa matricei x =
[[3 8]
 [9 5]]
Inversa matricei y=
[[ 0.63636364 -0.27272727]
 [-0.09090909  0.18181818]]

```

Ex5_11

În acest exercițiu trebuie să analizați 2 coduri și fără a rula codurile trebuie să vă dați seama care va fi răspunsul:

a)

```

def suma (x=4, y=-2):
    return x+y

print(suma())

```

b)

```

def suma (x=4, y=-2):
    return x+y

print(suma(5, -5))

```

a) 2
b) 0

Ex5_12

În acest exercițiu trebuie să creăm un cod în care să cerem de la utilizator să introducă limita de jos și limita de sus. Din acest șir de numere trebuie să cream o listă nouă lista_mea = [] în care să se regăsească doar numerele multele cu 4 și non multiple cu 3. Deoarece suntem la tema despre funcții codul creat trebuie să se bazeze pe o funcție creată de noi. La soluții vor fi date două răspunsuri a) cod care nu se bazează pe funcție creată de programist b) cod care se bazează pe o funcție creată de programist. În ambele cazuri după rularea codului trebuie să obținem următorul răspuns:

```

Introdu limita de jos: -30
Introdu limita de sus: 23
['-28', '-20', '-16', '-8', '-4', '4', '8', '16', '20']

```

```

a = int(input("Introdu limita de jos:"))
b = int(input("Introdu limita de sus:"))
lista = list(range(a,b))
lista_mea=filter(lambda c:c%4==0 and c%3!=0, lista)
print(list(lista_mea))

def sortare(lista1):
    lista2 = filter(lambda c:c%4==0 and c%3!=0, lista1)
    print(list(lista2))
sortare(lista)

```



```

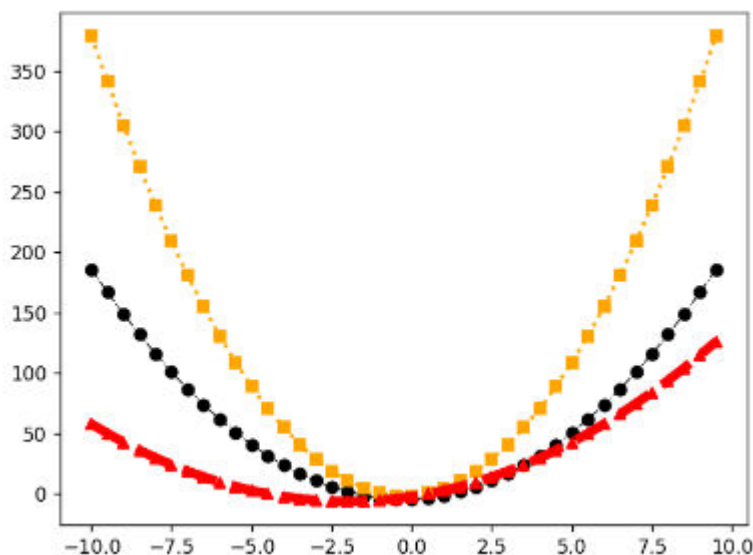
Introdu limita de jos:-30
Introdu limita de sus:23
[-28, -20, -16, -8, -4, 4, 8, 16, 20]
[-28, -20, -16, -8, -4, 4, 8, 16, 20]

```

Ex5_13

Folosind bibliotecile NumPy și Matplotlib creai în aceeași figură graficele următoarelor funcții $y = 2x^2 + x - 4$, $y = 4x^2 + 2x - 1$ și $y = x^2 + 4x - 2$. Liniile celor trei grafice trebuie să fie diferite: culoare, tip linie, grosime linie și marker linie.

După rularea codului un răspuns care putem să îl obținem poate arăta în felul următor:



```

import matplotlib.pyplot as myplt
import numpy as np

x = np.linspace(-20,20,num = 50)
def y1(x):
    return 2*x**2+x-4
def y2(x):
    return 4*x**2+2*x-1
def y3(x):
    return x**2+4*x-2

ylist1 = y1(x)
ylist2 = y2(x)
ylist3 = y3(x)
myplt.figure(1)
myplt.plot(x,ylist1,"-k", marker = "o")
myplt.plot(x,ylist2,"--y", marker = "s")
myplt.plot(x,ylist3,":r", marker = "^")
myplt.show()

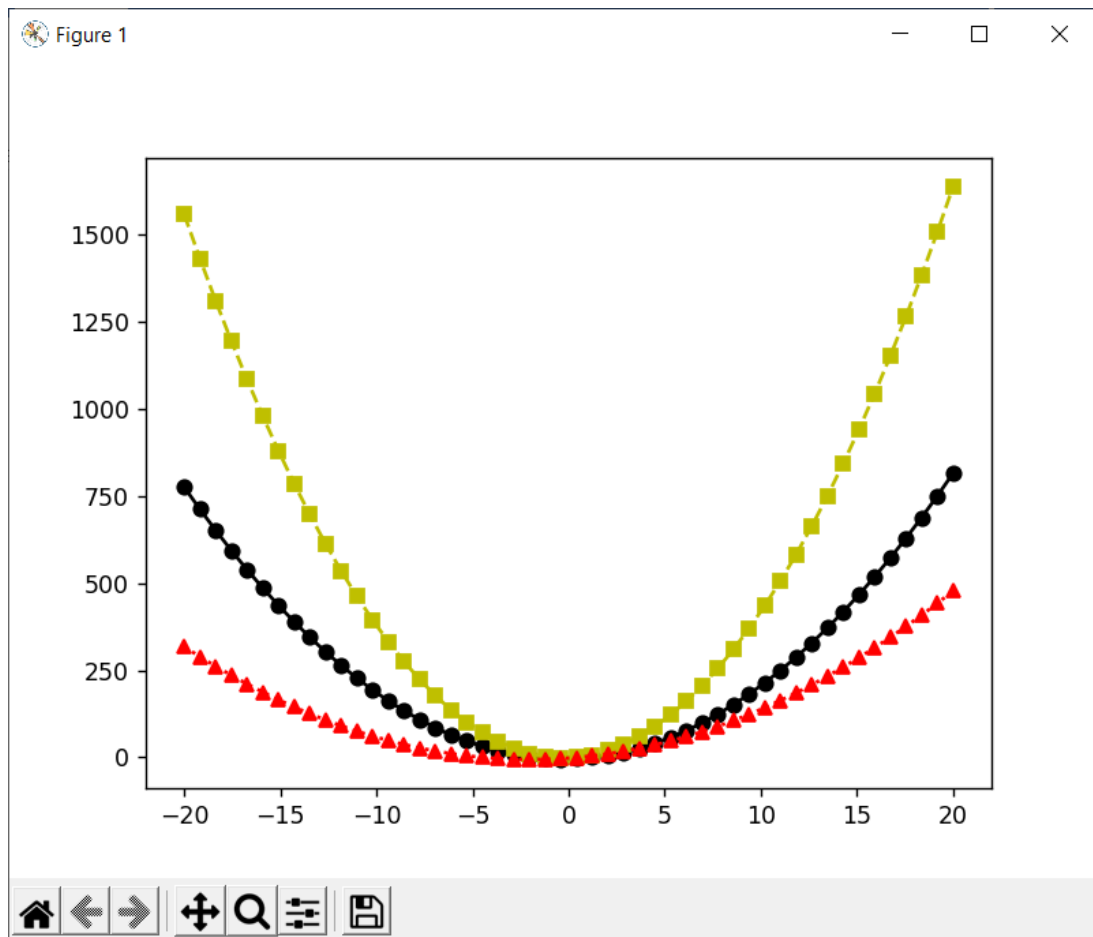
```

Mo	Coal	N.	Semnat	Data

GC Nr. 21-186 – Cojocari Dragoș

Coal

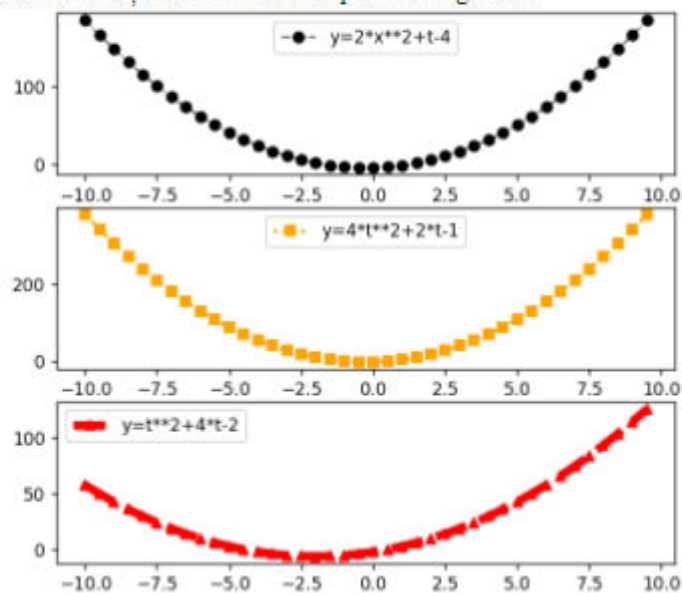
24



Ex5_14

Modificați codul din exemplul 5_13 astfel încât să obținem cele trei grafice ale ecuațiilor $y = 2x^2 + x - 4$, $y = 4x^2 + 2x - 1$ și $y = x^2 + 4x - 2$ în aceeași figură dar ca 3 sub figuri distincte.

După rulare trebuie să obținem următoarea reprezentare grafică:



Mo	Coal	N.	Semnat	Data

GC Nr. 21-186 – Cojocari Dragoș

Coal

25

```

import matplotlib.pyplot as myplt
import numpy as np

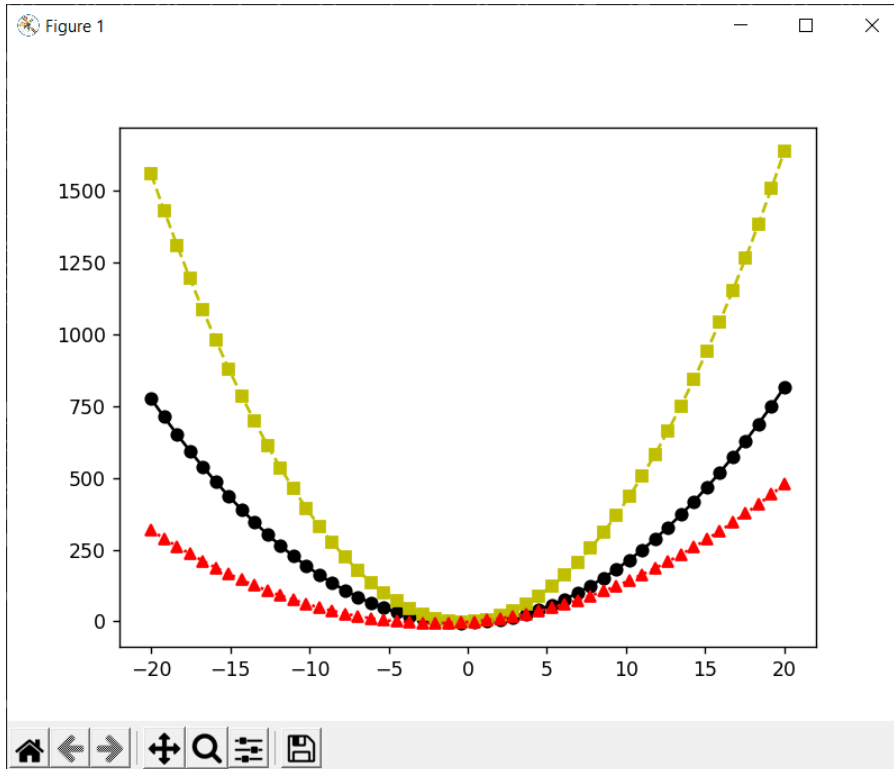
x = np.linspace(-10,10,num = 30)
def y1(x):
    return 2*x**2+x-4
def y2(x):
    return 4*x**2+2*x-1
def y3(x):
    return x**2+4*x-2

ylist1 = y1(x)
ylist2 = y2(x)
ylist3 = y3(x)

fig,(y1,y2,y3)=myplt.subplots(nrows=3,ncols=1)
y1.plot(x,ylist1,"-k",marker="o", label = '2*x**2+x-4')
y1.legend()
y2.plot(x,ylist2,"--y",marker="s", label = '4*x**2+2*x-1')
y2.legend()
y3.plot(x,ylist3,":r",marker="^", label = 'x**2+4*x-2')
y3.legend()

myplt.show()

```



Mo	Coal	N.	Semnat	Data

Exemple Turtle

Exemplu 1

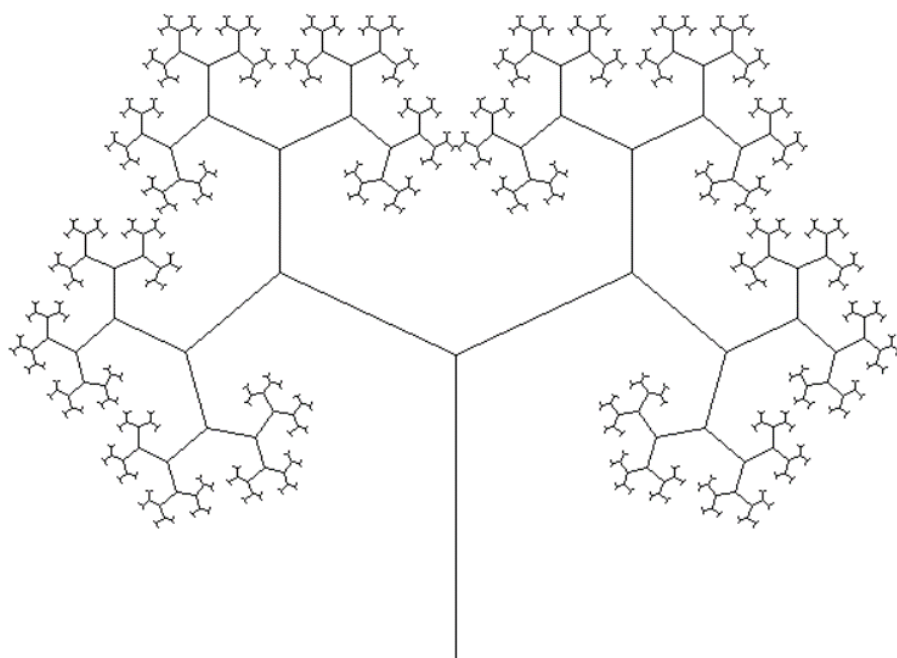
```
from turtle import Turtle, mainloop
from time import perf_counter as clock

def tree(plist, l, a, f):
    if l > 3:
        lst = []
        for p in plist:
            p.forward(l)
            q = p.clone()
            p.left(a)
            q.right(a)
            lst.append(p)
            lst.append(q)
        for x in tree(lst, l*f, a, f):
            yield None

def maketree():
    p = Turtle()
    p.setundobuffer(None)
    p.hideturtle()
    p.speed(0)
    p.getscreen().tracer(30,0)
    p.left(90)
    p.penup()
    p.forward(-210)
    p.pendown()
    t = tree([p], 200, 65, 0.6375)
    for x in t:
        pass

maketree()
mainloop()
```

					GC Nr. 21-186 – Cojocari Dragoş	Coal
Mo	Coal	N.	Semnat	Data		27



Exemplu 2

```
from turtle import *

def main():
    peacecolors = ("red3", "orange", "yellow",
                  "seagreen4", "orchid4",
                  "royalblue1", "dodgerblue4")

    reset()
    Screen()
    up()
    goto(-320,-195)
    width(70)

    for pcolor in peacecolors:
        color(pcolor)
        down()
        forward(640)
        up()
        backward(640)
        left(90)
        forward(66)
        right(90)
```

Mo	Coal	N.	Semnat	Data

```

width(25)
color("white")
goto(0,-170)
down()

circle(170)
left(90)
forward(340)
up()
left(180)
forward(170)
right(45)
down()
forward(170)
up()
backward(170)
left(90)
down()
forward(170)
up()

goto(0,300) # vanish if hideturtle() is not available ;-)
return "Done!"

if __name__ == "__main__":
    main()
    mainloop()

```



Exemplu 3

```
"""      turtle-example-suite:
```

```
        tdemo_wikipedia3.py
```

This example is
inspired by the Wikipedia article on turtle
graphics. (See example wikipedia1 for URLs)

First we create (ne-1) (i.e. 35 in this
example) copies of our first turtle p.
Then we let them perform their steps in
parallel.

Followed by a complete undo().

```
"""
```

```
from turtle import Screen, Turtle, mainloop
from time import perf_counter as clock, sleep
```

```
def mn_eck(p, ne,sz):
    turtlelist = [p]
    #create ne-1 additional turtles
```

```

for i in range(1,ne):
    q = p.clone()
    q.rt(360.0/ne)
    turtlelist.append(q)
    p = q
for i in range(ne):
    c = abs(ne/2.0-i)/(ne*.7)
    # let those ne turtles make a step
    # in parallel:
    for t in turtlelist:
        t.rt(360./ne)
        t.pencolor(1-c,0,c)
        t.fd(sz)

def main():
    s = Screen()
    s.bgcolor("black")
    p=Turtle()
    p.speed(0)
    p.hideturtle()
    p.pencolor("red")
    p.pensize(3)

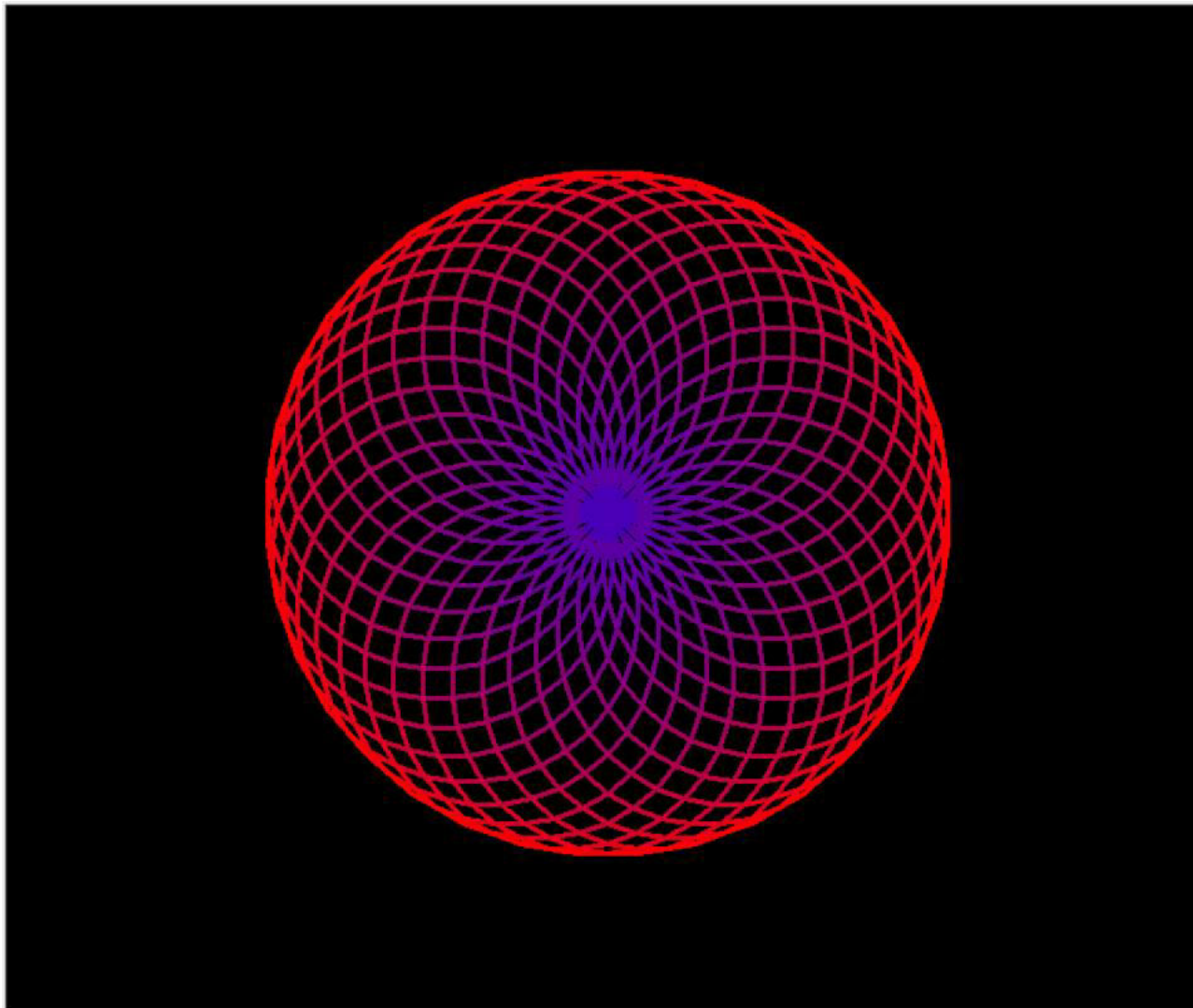
    s.tracer(36,0)

    at = clock()
    mn_eck(p, 36, 19)
    et = clock()
    z1 = et-at

    sleep(1)

main()
mainloop()

```

					GC Nr. 21-186 – Cojocari Dragoș	Coal
						32
Mo	Coal	N.	Semnat	Data		

Calculatorul

```

from tkinter import *
from tkinter import messagebox # mesaje
from tkinter import ttk #butoane

window=Tk()
window.title('Calculator')
butoane_calc=[
    "**(1/2)","(",")","**(","**", #**(1/2)-radical de ordinul 2
    "7","8","9","+","-", #**(1/-radical de ordinul n
    "4","5","6","*","/", #** x la puterea y
    "1","2","3","-/+","1/", #1/ --1/x=inversul lui x
    "0",".", "DELETE","=", "**2" #**2-ridicarea la patrat
]

r=1
c=0
calc_txt=Entry(window,width=25)
calc_txt.grid(row=0,column=0,columnspan=5)
def calc(key):
    global memory
    if key == "=":
        str1="**.+-(*)/0123456789"
        if calc_txt.get() [0] not in str1:
            calc_txt.insert(END,"Nu este cifra")
            messagebox.showerror("ERROR!!!Not Number")
        try:
            rezultat=eval(calc_txt.get())
            calc_txt.insert(END,"=" + str(rezultat))
        except:
            calc_txt.insert(END,"ERROR!!!")
            messagebox.showerror("Verifica valorile introduse")
    elif key == "DELETE":
        calc_txt.delete(0,END)
    elif key=="-/+":
        if "=" in calc_txt.get():
            calc_txt.delete(0,END)
        try:
            if calc_txt.get()[0]=="-":
                calc_txt.delete(0)
            else:
                calc_txt.insert(0,"-")
        except IndexError:
            pass
    else:
        if "=" in calc_txt.get():
            calc_txt.delete(0,END)
        calc_txt.insert(END,key)

```

```

for i in butoane_calc:
    rel=""
    cmd=lambda x=i:calc(x)
    ttk.Button(window,text=i,command=cmd).grid(row=r,column=c)
    c+=1
    if c>4:
        c=0
        r+=1

window.mainloop()

```

