Міністерство освіти і науки України

Державний університет ,,Житомирська політехніка”

Кафедра ІПЗ

Група: ВТ-21-1

Програмування мовою Python

Лабораторна робота №5

«Функції»

Виконав: Вигннч О. С.

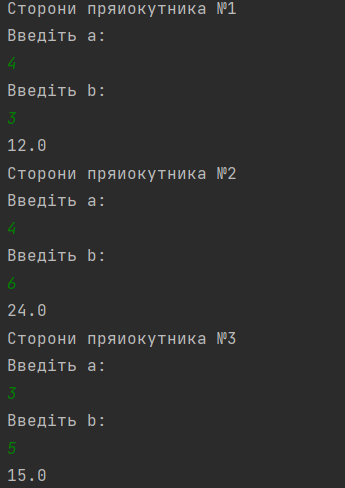
Прийняв: Морозов Д. С.

**Мета роботи:** ознайомитися методами роботи зі списками в мові Python

Виконання роботи:

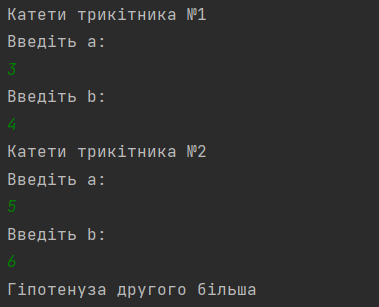
1. Користувач вводить дві сторони трьох прямокутників. Вивести їх площі.

def Area(a,b):  
 return a \* b;  
def Task1():  
 for i in range(3):  
 print(f"Сторони пряиокутника №{i + 1}")  
 print("Введіть а:")  
 a = float(input())  
 print("Введіть b:")  
 b = float(input())  
 print(Area(a, b))



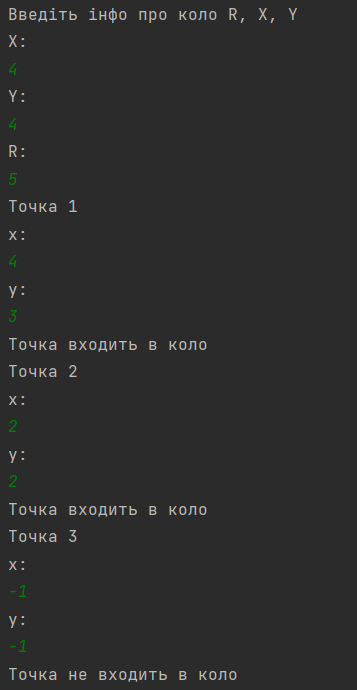
2. Дано катети двох прямокутних трикутників. Написати функцію обчислення довжини гіпотенузи цих трикутників. Порівняти і вивести яка з гіпотенуз більше, а яка менше.

def Hypotenuse(a,b):  
 return math.sqrt(a\*\*2 + b\*\*2)  
def Task2():  
 list = []  
 for i in range(2):  
 print(f"Катети трикітника №{i + 1}")  
 print("Введіть а:")  
 a = float(input())  
 print("Введіть b:")  
 b = float(input())  
 list.append(Hypotenuse(a, b))  
 if list[0] > list[1]:  
 print("Гіпотенуза першого більша")  
 else:  
 print("Гіпотенуза другого більша")



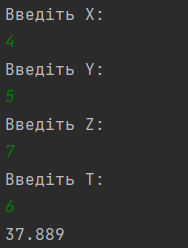
3. Задано коло (x-a)2 + (y-b)2 = R2 і точки Р (р1, р2), F (f1, f1), L (l1, l2). З'ясувати і вивести на екран, скільки точок лежить всередині кола. Перевірку, чи лежить точка всередині кола, оформити у вигляді функції.

def circleIncludes(a, b, r):  
 def Point(x, y):  
 if (x - a) \*\* 2 + (y - b) \*\* 2 < r \*\* 2:  
 print("Точка входить в коло")  
 else:  
 print('Точка не входить в коло')  
  
 for i in range(3):  
 print(f"Точка {i+ 1}")  
 print("x:")  
 x = int(input())  
 print("y:")  
 y = int(input())  
 Point(x, y)  
def Task3():  
 print("Введіть інфо про коло R, X, Y")  
 print("X:")  
 x = int(input())  
 print("Y:")  
 y = int(input())  
 print("R:")  
 r = int(input())  
 circleIncludes(x,y,r)



4. Дано числа X, Y, Z, Т - довжини сторін чотирикутника. Обчислити його площу, якщо кут між сторонами довжиною X і Y - прямий

def AreaFour(x,y,z,t):  
 Area1 = x \* y  
 hypoten = Hypotenuse(x,y)  
 p = (hypoten + z + t)/2  
 Area2 = math.sqrt(p\*(p-hypoten)\*(p-z)\*(p-t))  
 S = Area1 + Area2;  
 print(round(S, 3))  
def Task4():  
 print("Введіть X:")  
 x = float(input())  
 print("Введіть Y:")  
 y = float(input())  
 print("Введіть Z:")  
 z = float(input())  
 print("Введіть T:")  
 t = float(input())  
 AreaFour(x,y,z,t)



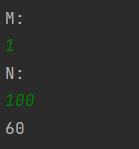
5. Знайти всі натуральні числа, що не перевищують заданого n, які діляться на кожне із заданих користувачем чисел

def Numbers(n, list):  
 for i in range(1,n + 1):  
 count = 0  
 for j in list:  
 if j % i == 0:  
 count += 1  
 if count == len(list):  
 print(i)  
list = [3,6,9]  
Numbers(7, list)



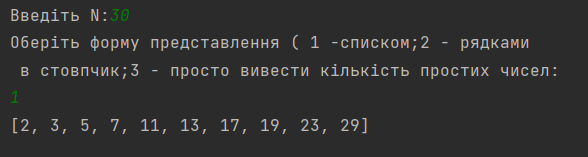
6. Скласти програму для знаходження чисел з інтервалу [М, N], що мають найбільшу кількість дільників.

def MostВivider(m,n):  
 list = []  
 for i in range(m,n+1):  
 count = 0  
 for j in range(1,i+1):  
 if i % j == 0:  
 count += 1  
 list.append(count)  
 maxValue = list.index(max(list)) + m  
 print(maxValue)  
def Task6():  
 print('M:')  
 m = int(input())  
 print('N:')  
 n = int(input())  
 MostВivider(m,n)



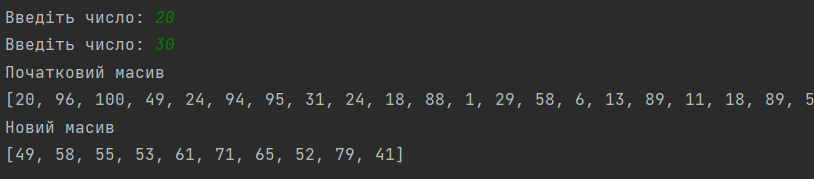
7. Написати функцію для пошуку всіх простих чисел від 0 до N з можливістю вибору формату представлення результату (списком; рядками в стовпчик; просто вивести кількість простих чисел.

def SimpleNumbers(n, v):  
 list = []  
 for i in range(1,n + 1):  
 count = 0  
 for j in range(1, i):  
 if i % j == 0:  
 count += 1  
 if count == 1:  
 list.append(i)  
 if v == 1:  
 print(list)  
 elif v == 2:  
 for i in list:  
 print(i)  
 else:  
 len(list)  
def Task7():  
 n = int(input("Введіть N:"))  
 print("Оберіть форму представлення ( 1 -списком;2 - рядками\n в стовпчик;3 - просто вивести кількість простих чисел:")  
 v = int(input())  
 SimpleNumbers(n,v)



8. Дано список з випадкових натуральних чисел довільної довжини. Написати програму, що формуватиме з заданого другий список, що міститиме тільки значення від MIN+bottom до MAX-upper. Де MIN і MAX – відповідно найменше і найбільше число в списку, а botton і upper – нижня і верхня межа значень вибірки нового списку. Програма має містити обробку винятків на випадок введення символів невірного типу, дробових чисел, вихід за межі мінімального і максимального значення.

def read\_int(text="Введіть число: "):  
 try:  
 value = int(input(text))  
 except Exception:  
 print("Не ціле число")  
 return read\_int(text)  
 else:  
 return value  
def Check(min, max):  
 x = read\_int()  
 if x > max or x < min:  
 print("Вийло за межі")  
 Check(min, max)  
 else: return x  
  
def Task8():  
 list = []  
 x = random.randint(0,100)  
 for i in range(1, x):  
 list.append(random.randint(0,100))  
 maxValue = max(list)  
 minValue = min(list)  
 upper = Check(minValue, maxValue)  
 bottom = Check(minValue, maxValue)  
 NewMax = maxValue - upper  
 NewMin = minValue + bottom  
 newList = []  
 for i in list:  
 if i > NewMin and i < NewMax:  
 newList.append(i)  
 print("Початковий масив")  
 print(list)  
 print("Новий масив")  
 print(newList)

Завдання 9. Для завдань 6 – 8 написати декоратор, що дозволить визначати час виконання програми. Виконати перевірку часу виконання написаних функцій для 10\*\*n елементів при n <=6 з кроком в n. Тобто визначити час виконання

def TimeOperation(func):  
 def wrapper():  
 t1 = time.perf\_counter()  
 func()  
 t2 = time.perf\_counter()  
 print(f"Часв виконання: {(t2-t1):.6f}")  
 return wrapper

@TimeOperation  
def Number10TK6():  
 MostВivider(1,10000)  
  
@TimeOperation  
def Number10TK7():  
 SimpleNumbers(10000,3)  
  
@TimeOperation  
def Number10TK8():  
 Task8()

