5.1 task 7

Дано:

Входные параметры:

Длина списка array1 и array2

Задание:



Выходные параметры:

Список array3

Дополнительные условия:

Использовать функцию map и анонимную функцию lambda

Текст программы:

from random import randint

def main():

length = int(input('Введите длину списков: '))

floor = int(input('Минимальный порог значений элементов: '))

ceiling = int(input('Максимальный порог значений элементов: '))

create = lambda length, floor, ceiling: [randint(floor, ceiling) for length in range(length)]

subt = lambda ar1, ar2: ar1 - ar2

array1, array2 = create(length, floor, ceiling), create(length, floor, ceiling)

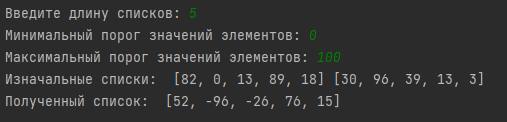
print('Изначальные списки: ', array1, array2)

array3 = list(map(subt, array1, array2))

print('Полученный список: ', array3)

main()

Результат тестирования:

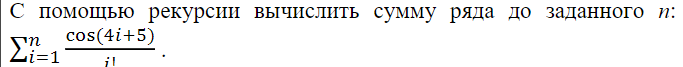


5.2 task 7

Дано:

Заданный n

Задание:



Выходные параметры:

Сумма ряда sum

Дополнительные условия:

Использовать рекурсивную функцию.

Текст программы:

from math import cos

from math import cos

def main():

n = int(input('Введите n: '))

i = 1

sum = 0

def fact(n):

if n == 0:

return 1

return fact(n - 1) \* n

def sum\_n(n, i, sum):

if i > n:

return sum

else:

return sum + sum\_n(n, i + 1, (cos(4\*i + 5)/fact(i)))

print(f'Сумма ряда до заданного n({n}): {"%.4f" % sum\_n(n, i, sum)}')

main()

Результат тестирования:





5.3 task 7

Дано:

Заданная точность эпсилон (e)

Задание:

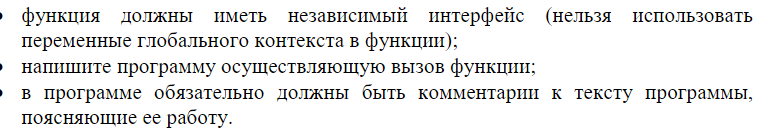
Реализовать функцию, вычисляющую рекурсивно сумму

Выходные параметры:

Значение суммы (sum)

Дополнительные условия:

Циклы использовать нельзя



Текст программы:

def main():

e = int(input('Введите 𝜀: ')) #задаваемая точность

n = 1

sum = 0

def fact(factExpr): #функция вычисления факториала

if factExpr == 0:

return 1

return fact(factExpr - 1) \* n

def sum\_n(e, n, sum): #рекурсивная функция

if n > e:

return sum

else:

return sum\_n(e, n + 1, sum + (((-1)\*\*(n - 1) \* n)/fact(2\*n))) #выводит е, увеличивает n на 1

#и увеличивает сумму

print(f'Сумма ряда c точностью 𝜀: {"%.4f" % sum\_n(e, n, sum)}') #вывод результата с вызовом функции sum\_n

main()

Результат:



