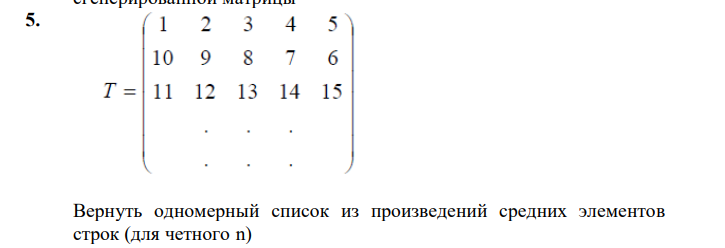
Л.р. №5 по ФЛП

Чередов Р.А., гр. ПИ-02

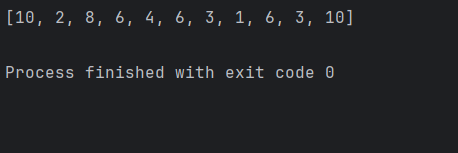
Задание №1



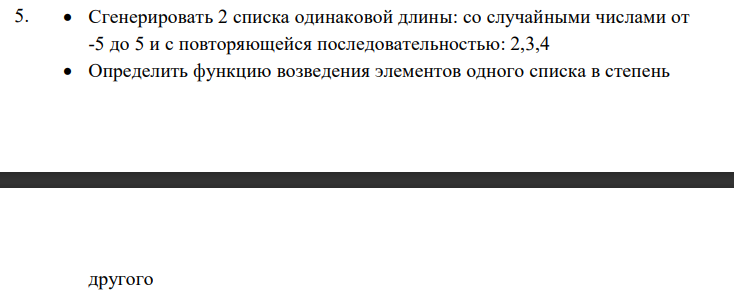
Реализация

n = 6# порядок матрицы  
  
matrix = [[i\*n + j + 1 if i % 2 == 0 else (i+1)\*n - j for j in range(n)] for i in range(n)]  
  
if len(matrix) != len(matrix[0]):  
 print("Матрица не является квадратной")  
else:  
 print("Матрица:")  
 for row in matrix:  
 print(row)  
  
 if n % 2 == 0:  
 result = [(lambda row: (row[len(row)//2] + row[len(row)//2-1])/2)(row) for row in matrix]  
 else:  
 result = [(lambda row: row[len(row)//2])(row) for row in matrix]  
 output = list(map(lambda x: x, result))  
 print("Результат:")  
 print(output)

Решение



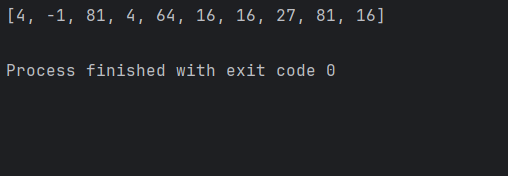
Задание №2



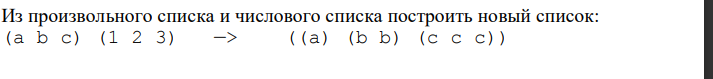
Реализация

import random  
  
# Генерация списка со случайными числами от -5 до 5  
random\_numbers = [random.randint(-5, 5) for \_ in range(10)]  
  
# Генерация списка с повторяющейся последовательностью  
repeating\_sequence = [2, 3, 4] \* 10  
  
# Определение функции для возведения элементов одного списка в степень другого  
def exponentiation(base, power):  
 return base \*\* power  
  
# Применение функции к спискам с использованием map и lambda  
result = list(map(lambda x, y: exponentiation(x, y), random\_numbers, repeating\_sequence))  
  
print(result)

Решение



Задание №3



Реализация

# Произвольный список  
letters = ['a', 'b', 'c']  
  
# Числовой список  
numbers = [1, 2, 3]  
  
# Использование функции map и zip  
mapped\_list = list(map(lambda x, y: [x] \* y, letters, numbers))  
  
# Использование спискового включения (list comprehension)  
list\_comprehension = [[x] \* y for x, y in zip(letters, numbers)]  
  
print("Mapped list:", mapped\_list)  
print("List comprehension:", list\_comprehension)

Решение

