Министерство науки и высшего образования Российской Федерации   
Федеральное государственное бюджетное образовательное   
учреждение высшего образования   
«Алтайский государственный технический   
университет им. И.И. Ползунова»

Факультет информационных технологий

Кафедра прикладной математики

Отчёт защищён с оценкой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Преподаватель Андреева А.Ю.

«\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2022 г.

Отчёт

по лабораторной работе № 5

**«СОЗДАНИЕ СПИСКОВ С ИСПОЛЬЗОВАНИЕ СПИСКОВЫХ ВКЛЮЧЕНИЙ И ПРОСТЕЙШАЯ ОБРАБОТКА С ИСПОЛЬЗОВАНИЕ ФУНКЦИЙ LAMBLA И MAP»**

по дисциплине «Функциональное и логическое программирование»

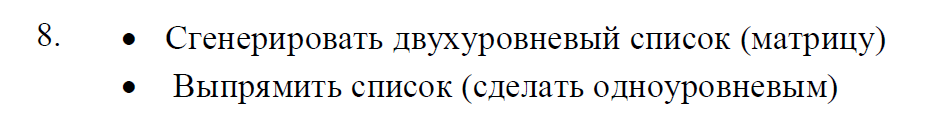
Студент группы ПИ-92 Савиков Д.А.

Преподаватель доцент, к.ф.-м.н. Андреева А.Ю.

Барнаул 2022

Номер в списке – 24

**Задание 1:** Используя списковые включения (list comprehension) сгенерировать одномерный список (возможно в 2-3 этапа). Выполнить обработку списка с использованием функции map и labmda.

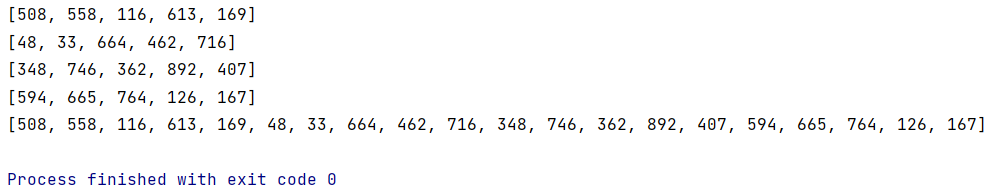


**Решение:**

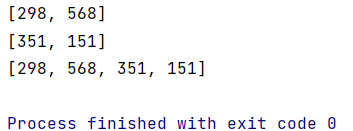
from random import randint  
  
def generate\_matrix(n, m):  
 *"""генерация двухуровневого списка"""* return [[randint(0, 1000) for j in range(m)] for i in range(n)]  
  
def matrix\_to\_list(matrix):  
 *"""выпрямление двухуровнего списка (превращение в одноуровневый)"""* return [x for row in matrix for x in row]  
  
def print\_matrix(matrix):  
 *"""красивый вывод двухуровневого списка"""* for row in matrix:  
 print(row)  
  
def main():  
 matrix = generate\_matrix(4, 5)  
 new\_list = matrix\_to\_list(matrix)  
  
 print\_matrix(matrix)  
 print(new\_list)  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

**Тесты:**

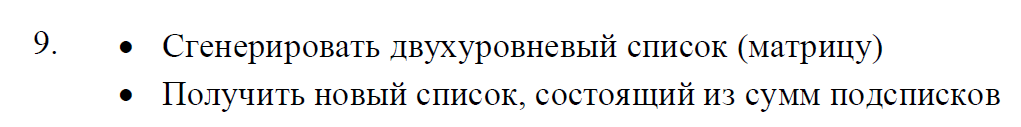
**1)**



2)



**Задание 2:** Используя списковые включения (list comprehension) сгенерировать двумерный список (возможно в 2-3 этапа) порядка n (для некоторых заданий порядок должен быть нечетным) и выполнить его обработку с использованием функции lambda.

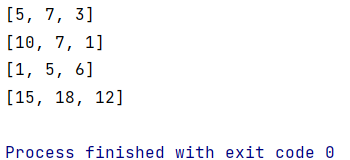


**Решение:**

from functools import reduce  
from random import randint  
  
def generate\_matrix(n, m):  
 *"""генерация двухуровневого списка"""* return [[randint(0, 10) for j in range(m)] for i in range(n)]  
  
def matrix\_to\_sum\_list(matrix):  
 *"""получение нового списока, состоящего из сумм подсписков"""* return [reduce(lambda x, y: x+y, row) for row in matrix]  
  
def print\_matrix(matrix):  
 *"""красивый вывод двухуровневого списка"""* for row in matrix:  
 print(row)  
  
def main():  
 matrix = generate\_matrix(3, 3)  
 new\_list = matrix\_to\_sum\_list(matrix)  
  
 print\_matrix(matrix)  
 print(new\_list)  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

**Тесты:**

1)



2)

