Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Уфимский государственный нефтяной технический университет»

Кафедра вычислительной техники и инженерной кибернетики

Задание по учебной практике №5

по дисциплине Информатика

«Динамические массивы данных»

Выполнил: студент гр. БПО09-24-02 Д.А. Хайруллин

Проверил: ст. преп. каф. ВТИК С.Г. Зайдуллина

Уфа 2025

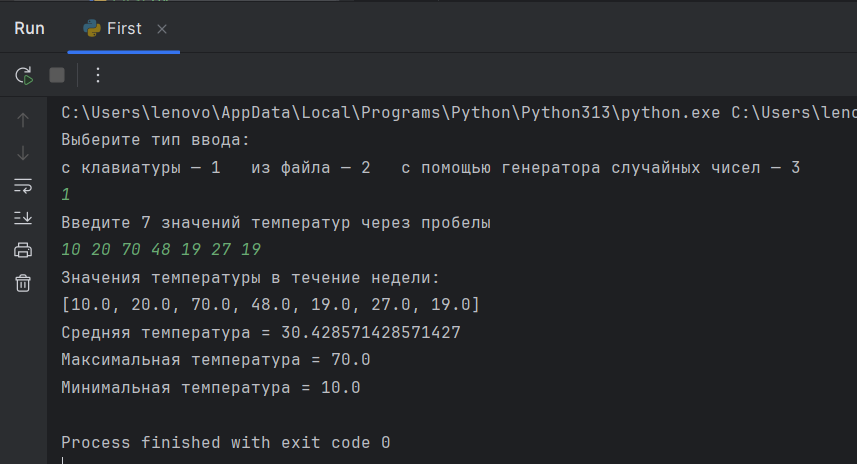
**Задание 1**

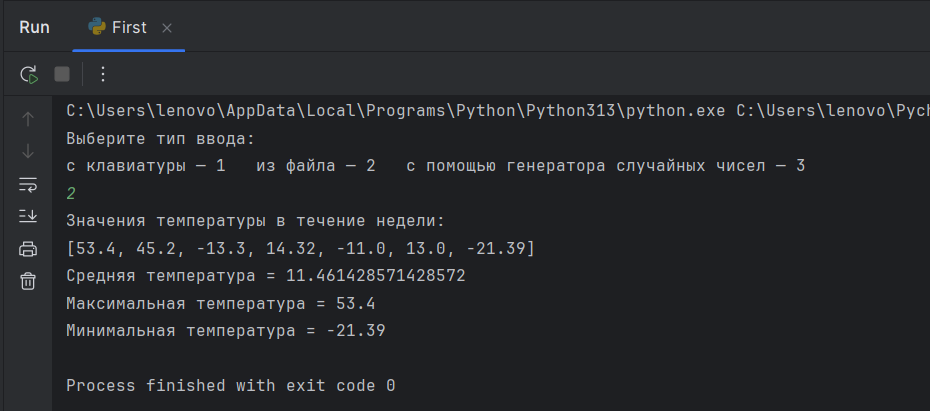
Написать программу, вычисляющую среднюю, максимальную, минимальную за неделю температуру воздуха (3 различные функции). Все ежедневные значения температур хранятся в одномерном массиве. Предусмотреть выбор ввода ежедневных температур с клавиатуры во время работы программы, из файла или генератором случайных чисел.

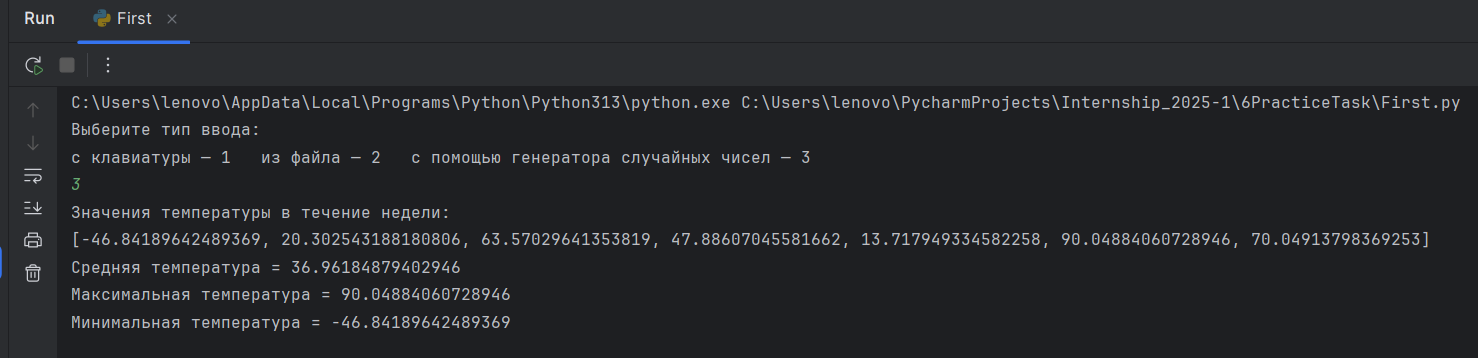
**Примечание:** как я понял, предполагалось, что задание мы должны были решать на языке С++, где нет max(), sum() и min() (в отличие ЯП Python). Поэтому в своём коде не буду их использовать.

**Код программы на Python**

*# 1. Написать программу, вычисляющую среднюю, максимальную, минимальную за неделю температуру воздуха (3 различные функции).  
# Все ежедневные значения температур хранятся в одномерном массиве.  
# Предусмотреть выбор ввода ежедневных температур с клавиатуры во время работы программы, из файла или генератором случайных чисел.*import random as rm  
def average\_temperature(temperature\_list: list):  
 summ = 0  
 for i in temperature\_list:  
 summ += i  
 averageTemperature = summ/ len(temperature\_list)  
 return averageTemperature  
  
def minimum\_temperature(temperature\_list: list):  
 min\_temp = 1000000000000000000000000  
 for i in temperature\_list:  
 if i < min\_temp:  
 min\_temp = i  
 return min\_temp  
  
def maximum\_temperature(temperature\_list: list):  
 max\_temp = 0  
 for i in temperature\_list:  
 if i > max\_temp:  
 max\_temp = i  
 return max\_temp  
choice = input("Выберите тип ввода:\nс клавиатуры — 1 из файла — 2 с помощью генератора случайных чисел — 3\n")  
if choice != 1 and choice != 2 and choice != 3:  
 raise print("Перезапустите программу и введите цифру 1, 2 или 3" )  
match choice:  
 case "1":  
 temperature\_str = input("Введите 7 значений температур через пробелы\n")  
 temperature\_list\_str= temperature\_str.split(sep = " ")  
 temperature\_l1st = list(map(float , temperature\_list\_str))  
 case "2":  
 with open("Read.txt", "r", encoding="utf-8") as file:  
 temperature\_str = file.read()  
 temperature\_list\_str = temperature\_str.split(sep=" ")  
 temperature\_l1st = list(map(float, temperature\_list\_str))  
 case "3":  
 temperature\_l1st = [rm.uniform(-100.0, 100.0) for \_ in range(7)]  
print(f"Значения температуры в течение недели:\n{temperature\_l1st}")  
print(f"Cредняя температура = {average\_temperature(temperature\_l1st)}")  
print(f"Максимальная температура = {maximum\_temperature(temperature\_l1st)}")  
print(f"Минимальная температура = {minimum\_temperature(temperature\_l1st)}")  
  
  
**Результаты:**

****

****

****

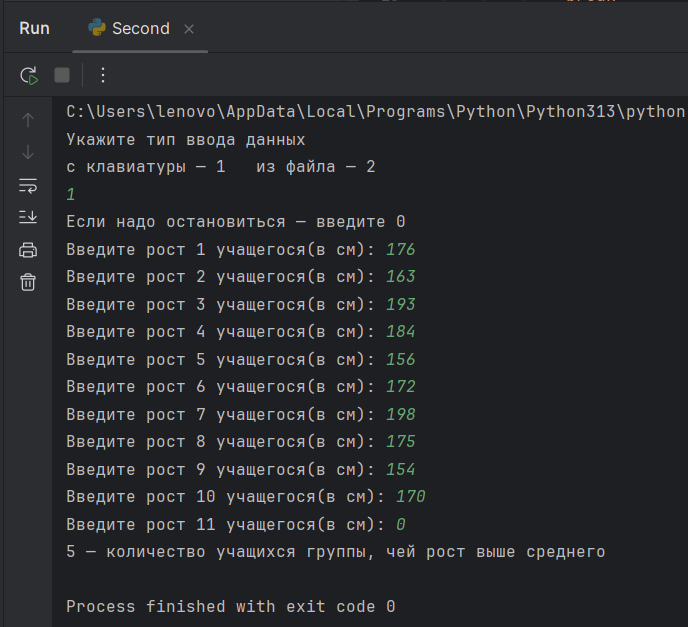
**Задание 2**

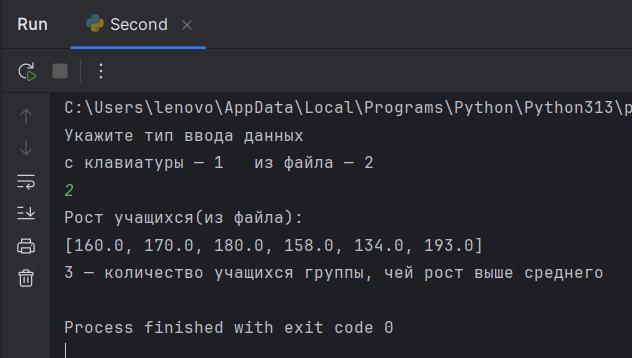
Написать программу, вычисляющую количество учащихся группы, чей рост выше среднего. Рост вводится в массив данных в процессе работы программы, последовательность заканчивается значением 0. Предусмотреть выбор ввода данных с клавиатуры или из файла.

**Код программы на Python**

*# Написать программу, вычисляющую количество учащихся группы, чей рост выше среднего.  
# Рост вводится в массив данных в процессе работы программы, последовательность заканчивается значением 0.  
# Предусмотреть выбор ввода данных с клавиатуры или из файла.*choice = int(input("Укажите тип ввода данных\nс клавиатуры — 1 из файла — 2\n"))  
if choice != 1 and choice != 2:  
 raise print("Перезапустите программу и введите цифру 1 или 2" )  
if choice == 1:  
 i = 1  
 height\_of\_students = []  
 print("Если надо остановиться — введите 0")  
 while True:  
 height\_of\_student = float(input(f"Введите рост {i} учащегося(в см): "))  
 if height\_of\_student == 0.0:  
 break  
 height\_of\_students.append(height\_of\_student)  
 i += 1  
  
else:  
 with open("Read+.txt", "r", encoding="utf-8") as file:  
 height\_str = file.read()  
 height\_list\_str = height\_str.split(sep=" ")  
 height\_of\_students = list(map(float, height\_list\_str))  
 print(f"Рост учащихся(из файла):\n{height\_of\_students}")  
  
average\_height = sum(height\_of\_students) / len(height\_of\_students)  
c0unt = 0  
for h in height\_of\_students:  
 if h > average\_height:  
 c0unt += 1  
  
print(f"{c0unt} — количество учащихся группы, чей рост выше среднего")

**Результаты:**

****

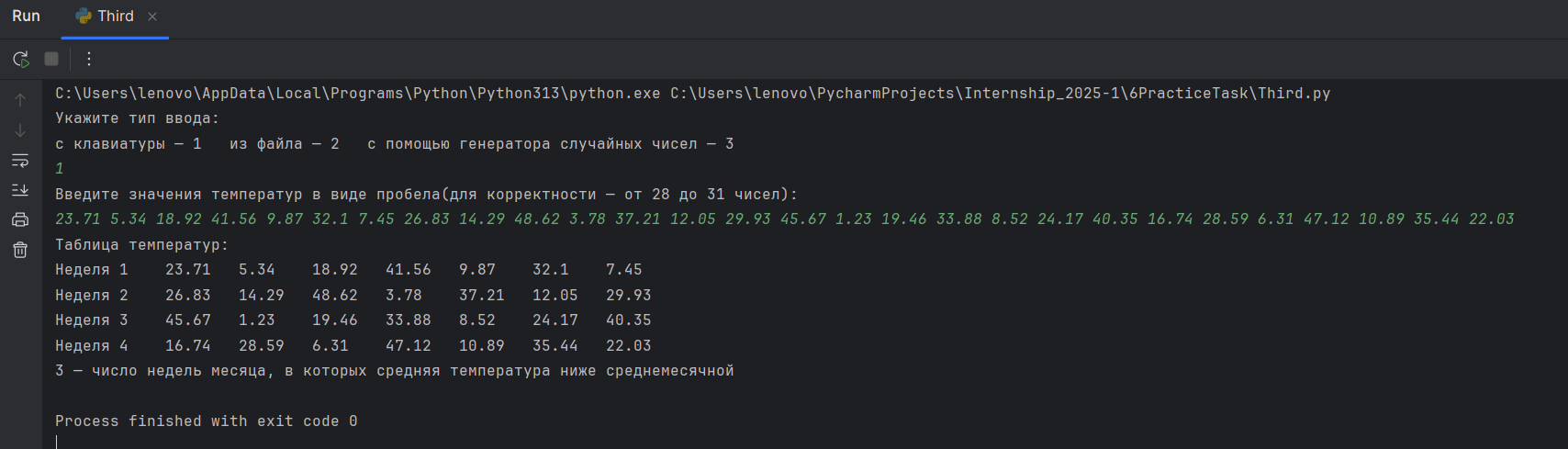
****

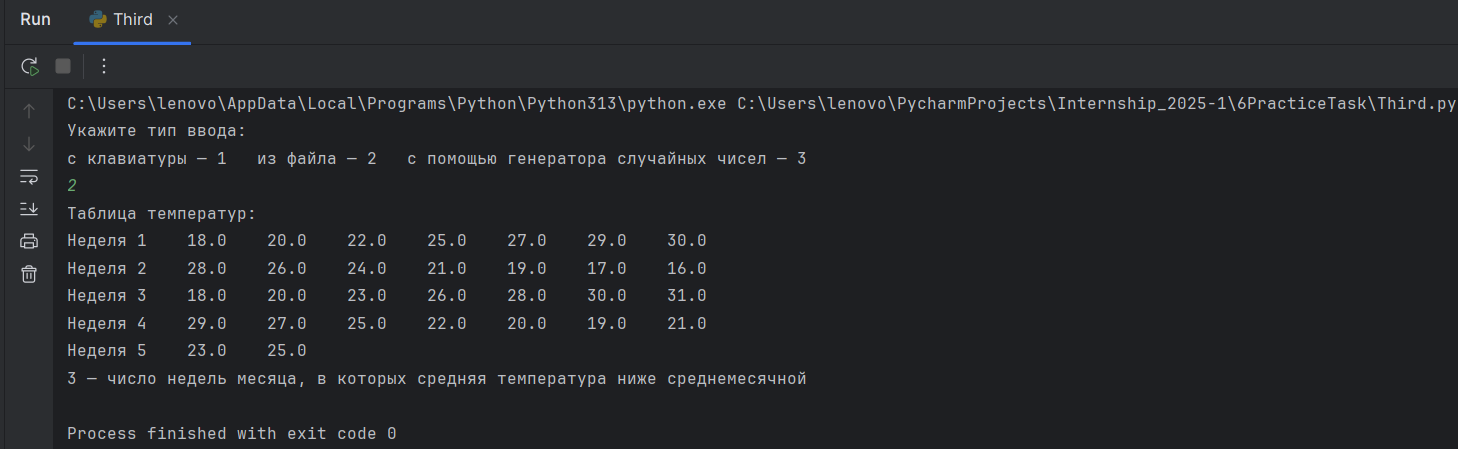
**Задание 3**

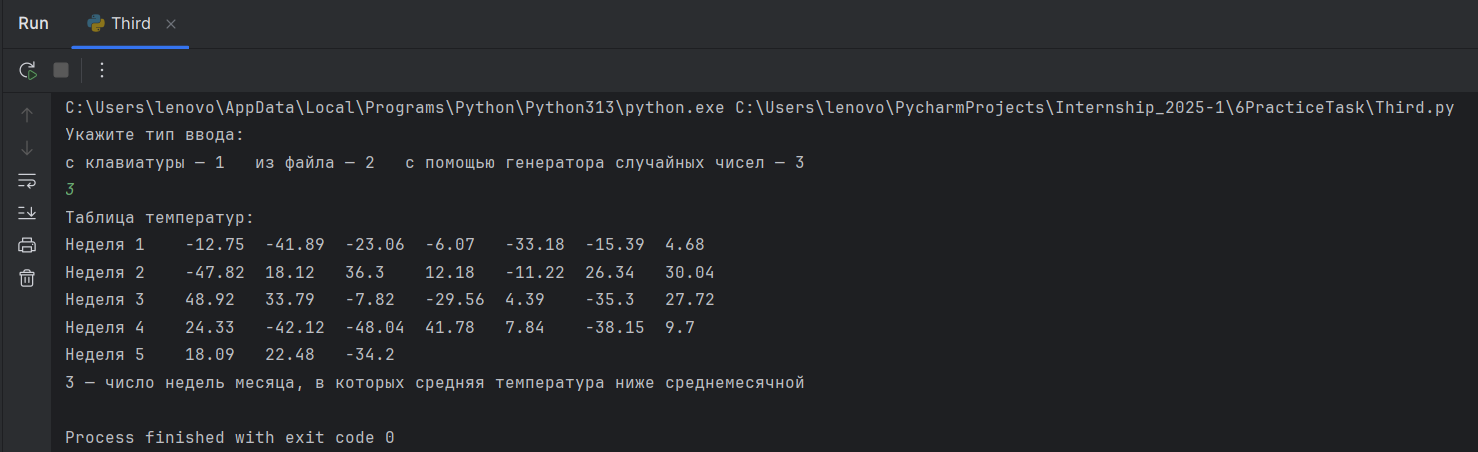
Написать программу, вычисляющую недели месяца, в которых средняя температура ниже среднемесячной. Все ежедневные значения температур хранятся в двумерном массиве. Предусмотреть выбор ввода ежедневных температур с клавиатуры во время работы программы, из файла или генератором случайных чисел.

**Код программы на Python**

*# Написать программу, вычисляющую недели месяца, в которых средняя температура ниже среднемесячной.  
# Все ежедневные значения температур хранятся в двумерном массиве.  
# Предусмотреть выбор ввода ежедневных температур с клавиатуры во время работы программы, из файла или генератором случайных чисел.*import random as rm  
  
choice = int(input("Укажите тип ввода:\nс клавиатуры — 1 из файла — 2 с помощью генератора случайных чисел — 3\n"))  
str\_temperature: list[str]  
temperature\_list: list[float]  
month\_temperature = []  
if choice != 1 and choice != 2 and choice != 3:  
 raise print("Перезапустите программу и введите цифру 1, 2 или 3" )  
match choice:  
 case 1:  
 raw\_str = input("Введите значения температур в виде пробела(для корректности — от 28 до 31 чисел):\n")  
 str\_temperature = raw\_str.split(" ")  
 temperature\_list = list(map(float, str\_temperature))  
 case 2:  
 with open ("Read++.txt", "r", encoding="utf-8") as file:  
 raw\_str = file.read()  
 str\_temperature = raw\_str.split(" ")  
 temperature\_list = list(map(float, str\_temperature))  
 case 3:  
 temperature\_list = [round(rm.uniform(-50.0, 50.0),2) for day in range(rm.randint(28, 31))]  
 str\_temperature = temperature\_list  
  
if len(str\_temperature) % 7 == 0:  
 weeks\_number = len(str\_temperature)//7  
else:  
 weeks\_number = (len(str\_temperature)//7)+1  
average\_temperature\_month = sum(temperature\_list)/len(temperature\_list)  
for i in range(weeks\_number):  
 c0unt = 0  
 week\_temperature = []  
 if len(temperature\_list) < 7 and len(temperature\_list)-(len(temperature\_list)//7)\*7 < 7:  
 for k in range(len(temperature\_list)-(len(temperature\_list)//7)\*7):  
 week\_temperature.append(temperature\_list[k])  
 else:  
 for k in range(7):  
 week\_temperature.append(temperature\_list[k])  
 month\_temperature.append(week\_temperature)  
 temperature\_list = temperature\_list[7:]  
print("Taблица температур:")  
for number\_of\_the\_week, week in enumerate(month\_temperature):  
 print(f"Неделя {number\_of\_the\_week+1}\t", end="")  
 for day\_temperature in week:  
 print(f"{day\_temperature}\t", end="")  
 print("\n", end="")  
  
  
week\_c0unt = 0  
for week in month\_temperature:  
 average\_temperature\_week = sum(week)/ len(week)  
 if average\_temperature\_month < average\_temperature\_week:  
 week\_c0unt += 1  
print(f"{week\_c0unt} — число недель месяца, в которых средняя температура ниже среднемесячной")  
  
**Результаты:**





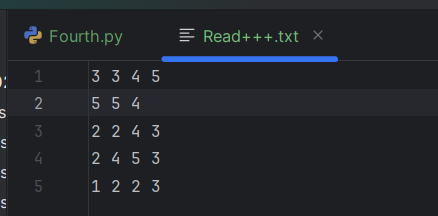


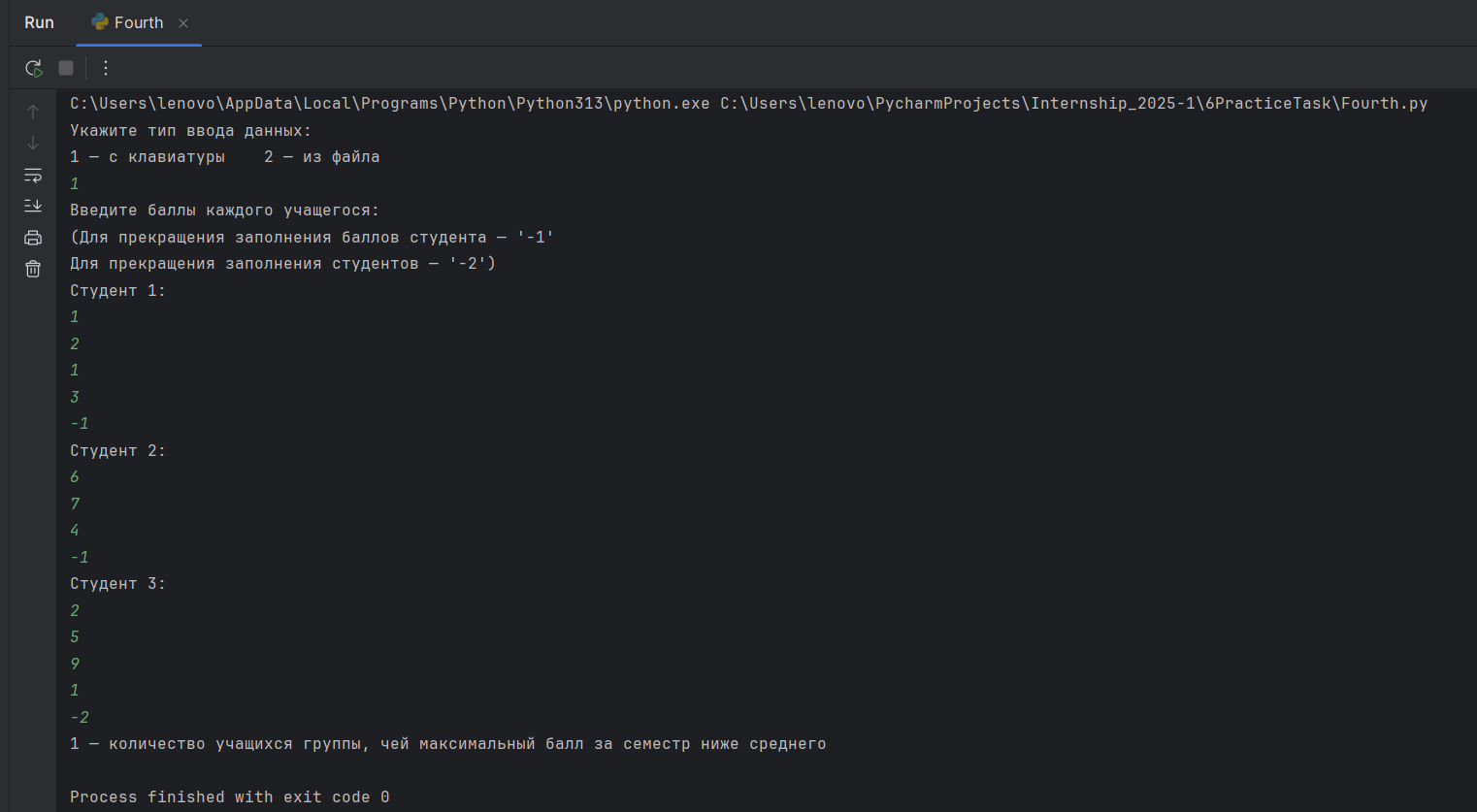
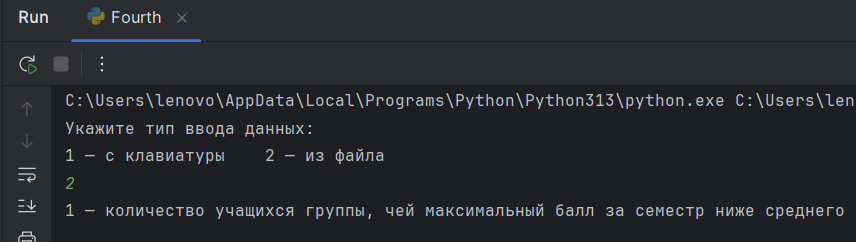
**Задание 4**

Написать программу, вычисляющую количество учащихся группы, чей максимальный балл за семестр ниже среднего. Успеваемость вводится в массив данных в процессе работы программы в виде последовательности оценок каждого студента, последовательность заканчивается значением -1. Формирование массива заканчивается -2. Предусмотреть выбор ввода данных с клавиатуры или из файла.

**Код программы на Python**

*# Написать программу, вычисляющую количество учащихся группы, чей максимальный балл за семестр ниже среднего.  
# Успеваемость вводится в массив данных в процессе работы программы в виде последовательности оценок каждого студента,  
# последовательность заканчивается значением -1. Формирование массива заканчивается -2.  
# Предусмотреть выбор ввода данных с клавиатуры или из файла.*students\_list = []  
choice = int(input("Укажите тип ввода данных:\n1 — с клавиатуры 2 — из файла\n"))  
if choice != 1 and choice != 2:  
 raise print("Перезапустите программу и введите цифру 1 или 2" )  
match choice:  
 case 1:  
 print("Введите баллы каждого учащегося:\n(Для прекращения заполнения баллов студента — '-1'\nДля прекращения заполнения студентов — '-2')")  
 students\_number = 1  
 while True:  
 print(f"Студент {students\_number}: ")  
 students\_number += 1  
 students\_scores = []  
 while True:  
 score = int(input())  
 students\_scores.append(score)  
 if score == -1 or score == -2:  
 students\_scores = students\_scores[:-1]  
 break  
 students\_list.append(students\_scores)  
 if score == -2:  
 break  
 case 2:  
 with open ("Read+++.txt", "r+", encoding="utf-8") as file:  
 line = "none"  
 while line != "":  
 line = file.readline()  
 if line == "":  
 break  
 line = list(map(int, line.split(sep = " ")))  
 students\_list.append(line)  
  
  
amount\_of\_points = 0  
scores\_count = 0  
list\_of\_maximum\_students\_scores = []  
for \_ in students\_list:  
 students\_amount\_of\_points = sum(\_)  
 students\_scores\_count = len(\_)  
 list\_of\_maximum\_students\_scores.append(max(\_))  
 amount\_of\_points += students\_amount\_of\_points  
 scores\_count += students\_scores\_count  
  
average\_score = amount\_of\_points/scores\_count  
below\_average\_count = len([x for x in list\_of\_maximum\_students\_scores if x < average\_score])  
print(f"{below\_average\_count} — количество учащихся группы, чей максимальный балл за семестр ниже среднего")  
  
**Результаты:**



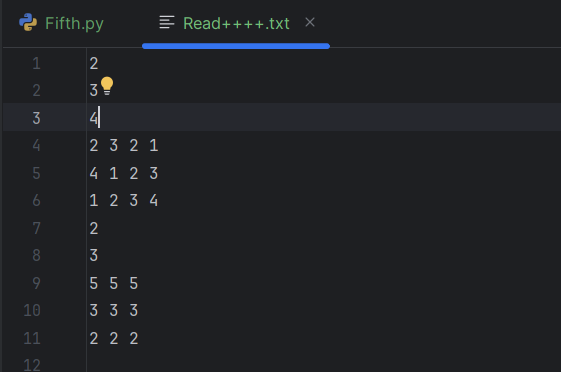
**Задание 5**

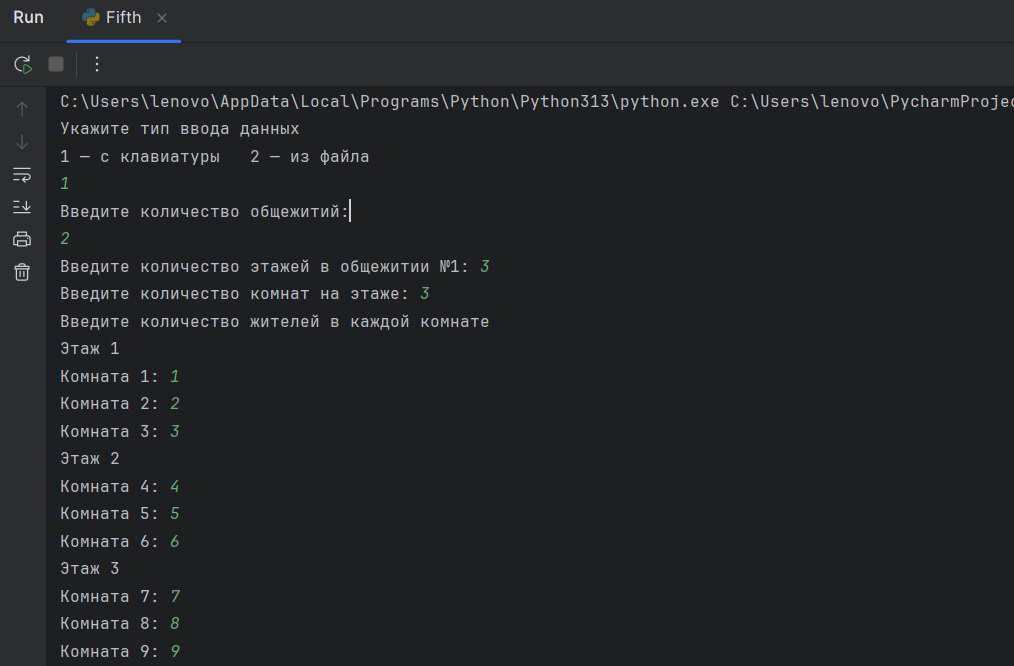
Написать программу, которая возвращает номер общежития, в котором есть наиболее заселенный этаж. Входные данные (число проживающих в комнате) находятся в трехмерном массиве M[i][j][k], где i - номер общежития, j - номер этажа, k - номер комнаты. Предусмотреть выбор ввода данных с клавиатуры или из файла.

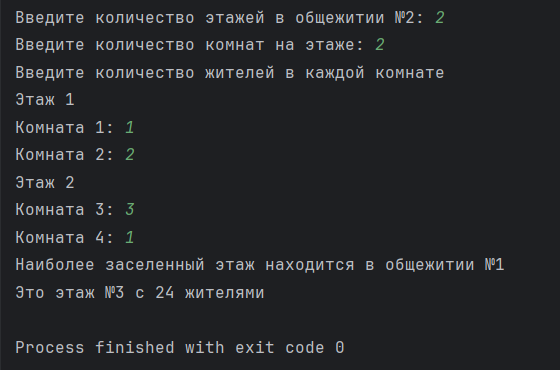
**Код программы на Python**

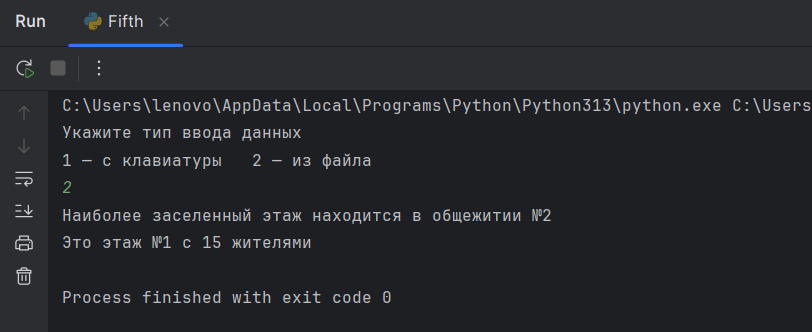
*# Написать программу, которая возвращает номер общежития, в котором есть наиболее заселенный этаж.  
# Входные данные (число проживающих в комнате) находятся в трехмерном массиве M[i][j][k],  
# где i - номер общежития, j - номер этажа, k - номер комнаты.  
# Предусмотреть выбор ввода данных с клавиатуры или из файла.*choice = int(input("Укажите тип ввода данных\n1 — с клавиатуры 2 — из файла\n"))  
if choice != 1 and choice != 2:  
 raise print("Перезапустите программу и введите цифру 1 или 2")  
array\_of\_dormitories = []  
match choice:  
 case 1:  
 number\_of\_dormitories = int(input("Введите количество общежитий:\n"))  
 dormitories\_number = 1  
 for \_ in range(number\_of\_dormitories):  
 number\_of\_floors = int(input(f"Введите количество этажей в общежитии №{dormitories\_number}: "))  
 number\_of\_rooms\_on\_the\_floor = int(input("Введите количество комнат на этаже: "))  
 number\_of\_rooms = number\_of\_rooms\_on\_the\_floor \* number\_of\_floors  
 print("Введите количество жителей в каждой комнате")  
 rooms\_number = 1  
 people\_in\_dormitories = []  
 floors\_number = 1  
 for floor in range(number\_of\_floors):  
 print(f"Этаж {floors\_number}")  
 people\_in\_floor = []  
 for room in range(number\_of\_rooms\_on\_the\_floor):  
 people\_in\_room = int(input(f"Комната {rooms\_number}: "))  
 people\_in\_floor.append(people\_in\_room)  
 rooms\_number += 1  
 people\_in\_dormitories.append(people\_in\_floor)  
 floors\_number += 1  
 array\_of\_dormitories.append(people\_in\_dormitories)  
 dormitories\_number += 1  
  
 case 2:  
 with open("Read++++.txt", "r", encoding="utf-8") as file:  
 number\_of\_dormitories = int(file.readline())  
 for \_ in range(number\_of\_dormitories):  
 number\_of\_floors = int(file.readline())  
 number\_of\_rooms\_on\_the\_floor = int(file.readline())  
 people\_in\_dormitories = []  
 for \_ in range(number\_of\_floors):  
 people\_in\_floor = list(map(int, file.readline().split()))  
 people\_in\_dormitories.append(people\_in\_floor)  
 array\_of\_dormitories.append(people\_in\_dormitories)  
  
  
max\_residents = -1  
result\_dormitory = -1  
result\_floor = -1  
  
for dormitory\_idx, dormitory in enumerate(array\_of\_dormitories, 1):  
 for floor\_idx, floor in enumerate(dormitory, 1):  
 total\_residents = sum(floor)  
 if total\_residents > max\_residents:  
 max\_residents = total\_residents  
 result\_dormitory = dormitory\_idx  
 result\_floor = floor\_idx  
  
print(f"Наиболее заселенный этаж находится в общежитии №{result\_dormitory}")  
print(f"Это этаж №{result\_floor} с {max\_residents} жителями")

**Результаты:**









**Выводы по работе.**

В ходе выполнения лабораторной работы я закрепил знания о динамических массивах данных в рамках языка Python.