Высшее государственное образовательное учреждение

«Санкт-Петербургский Государственный Морской Технический Университет»

Лабораторная работа №1 по программированию

Работу выполнил

Студент 1 курса факультета

цифровых промышленных технологий

группы 20121 Красноперов Данил

Проверил Поделенюк Павел Петрович

Санкт-Петербург

2023

# 1.Цели работы

При работе над данной лабораторной работой будут затронуты следующие вопросы:

1. Работа с файлами,
2. Работа с несколькими функциями, а не только с main,
3. Реализации рекурсивной функции.

# 2.Ход работы

1. Создадим файл Input.txt, запишем в него вводные данные и считаем их.
2. Создадим функцию «find\_solution», которая принимает список чисел и общую сумму.
3. Определим рекурсивную функцию «recursion», которая принимает индекс i текущего числа, текущую сумму current\_sum и строку a, представляющую текущее выражение.
4. Внутри функции «recursion» произведем проверку, равен ли индекс i общему количеству чисел n. Если да, то происходит проверка на равенство текущей суммы current\_sum и общей суммы total\_sum. В случае успеха возвращается строка с выражением.
5. Если проверка условия (пункт 4) не выполняется, то вызываются варианты со знаками "+" и "-", используя текущий индекс, обновленную сумму и строку с выражением.
6. Затем в основной функции «find\_solution» вызывается рекурсивная функция «recursion» с начальными параметрами исходным числом и начальным выражением.
7. После этого функция «find\_solution» возвращает результат, если результат не найден, то возвращает строку "no solution".
8. Запишем наше решение в файл Output.txt.

# 3.Результат работы

Наш файл Input.txt выглядит следующим образом:

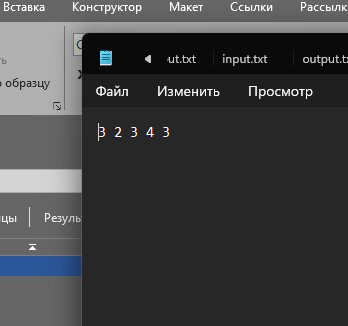


Рисунок 1 Файл Input.txt

После его выполнения в папке с программой появляется файл Output.txt, содержащий наше решение

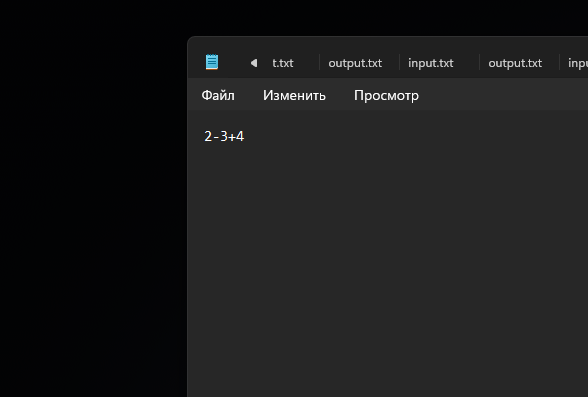
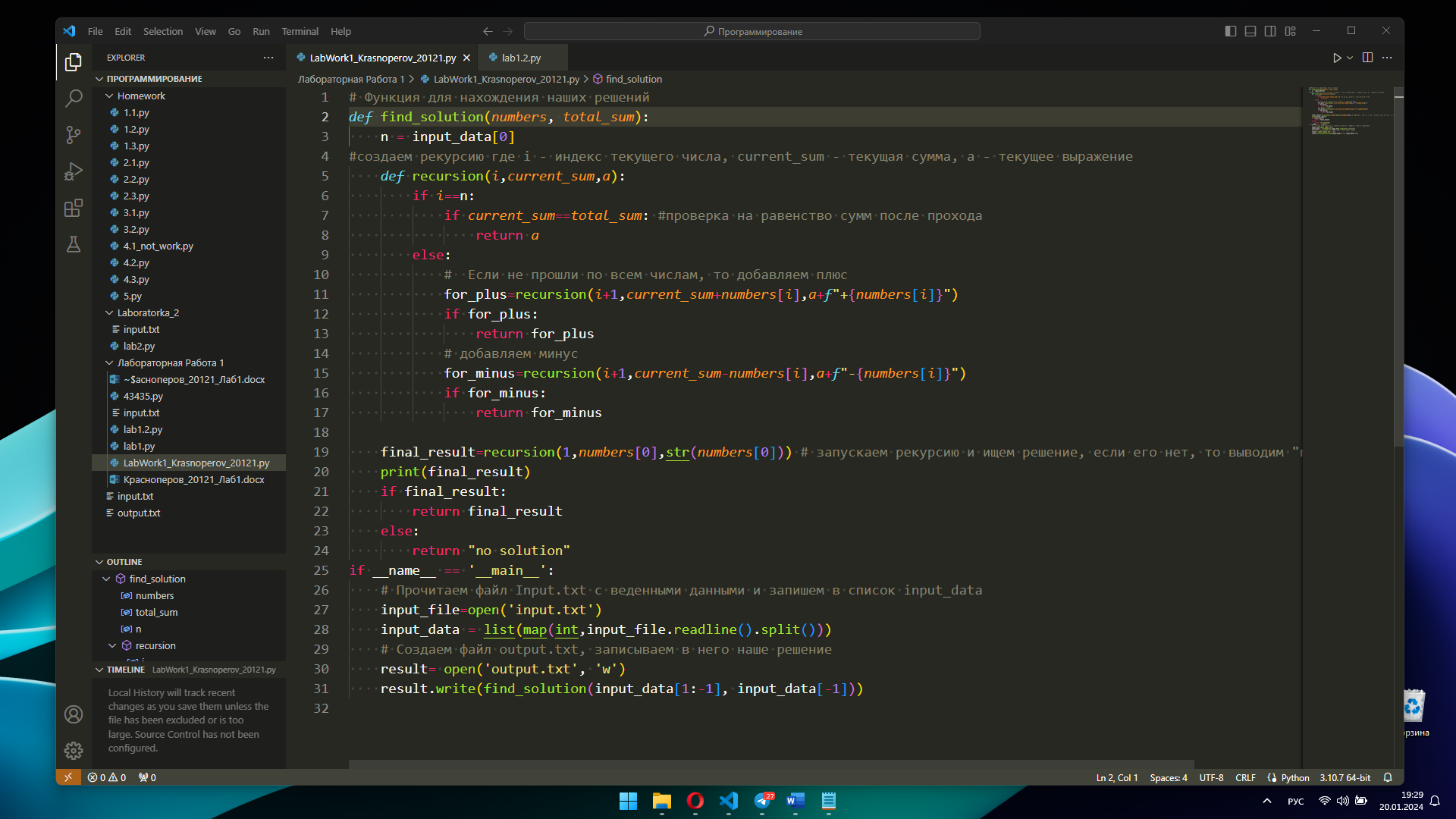


Рисунок 2Файл Output.txt

# 4.Вывод

При работе над данной лабораторной работой я получил навыки работы с файлами, несколькими функциями и реализации рекурсивной функции в языке программирования Python.

# 5.Листинг кода



# Функция для нахождения наших решений

*def* find\_solution(*numbers*, *total\_sum*):

    n = input\_data[0]

#создаем рекурсию где i - индекс текущего числа, current\_sum - текущая сумма, а - текущее выражение

*def* recursion(*i*,*current\_sum*,*a*):

        if *i*==n:

            if *current\_sum*==*total\_sum*: #проверка на равенство сумм после прохода

                return *a*

        else:

            #  Если не прошли по всем числам, то добавляем плюс

            for\_plus=recursion(*i*+1,*current\_sum*+*numbers*[*i*],*a*+*f*"+{*numbers*[*i*]}")

            if for\_plus:

                return for\_plus

            # добавляем минус

            for\_minus=recursion(*i*+1,*current\_sum*-*numbers*[*i*],*a*+*f*"-{*numbers*[*i*]}")

            if for\_minus:

                return for\_minus

    final\_result=recursion(1,*numbers*[0],str(*numbers*[0])) # запускаем рекурсию и ищем решение, если его нет, то выводим "no solutuon"

    print(final\_result)

    if final\_result:

        return final\_result

    else:

        return "no solution"

if \_\_name\_\_ == '\_\_main\_\_':

    # Прочитаем файл Input.txt c веденными данными и запишем в список input\_data

    input\_file=open('input.txt')

    input\_data = list(map(int,input\_file.readline().split()))

    # Создаем файл output.txt, записываем в него наше решение

    result= open('output.txt', 'w')

    result.write(find\_solution(input\_data[1:-1], input\_data[-1]))