САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

ФАКУЛЬТЕТ ИНФОКОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ

Отчет по лабораторной работе 1

по курсу «Алгоритмы и структуры данных»

Тема: Сортировка вставками, выбором, пузырьковая

Вариант: 21

Выполнила:

Сухорукова Ю.Е.

К3124

Проверила:

Артамонова В.Е.

Санкт-Петербург

2022 г.

Содержание отчета

**Содержание отчета**

**Задачи по варианту**

Задача 1. Название задачи [N баллов]

**Вывод**

**Задачи по варианту**

**Задача 1. Сортировка вставкой**

Используя код процедуры Insertion-sort, напишите программу и проверьте сортировку массива A = {31, 41, 59, 26, 41, 58}

**Формат входного файла(input.txt).** В первой строке входного файла содержится число n(1 <= n <= 10 ^ 3) – число элементов в массиве. Во второй строке находятся n различных целых чисел, по модулю не превосходящих 10 ^ 9.

**Формат выходного файла(output.txt).** Одна строка выходного файла с отсортированным массивом. Между любыми двумя числами должен стоять ровно один пробел.

Ограничение по времени. 2 сек.

Ограничение по памяти. 256 мб.

Выберите любой набор данных, подходящих по формату, и протестируйте алгоритм.

Листинг кода.

import time

import tracemalloc

tracemalloc.start()

start\_time = time.time()

input = open(‘input.txt’, ‘r’)

output = open(‘output.txt’, ‘w’)

n = int(input.readline() [0])

a = [int(x) for x in input.readline().split()]

for i in range(0, n):

k = a[i]

j = i – 1

while j >= 0 and a[j] > k:

a[j+1] = a[j]

j -= 1

a[j+1] = k

for i in a:

output.write(str(i))

output.write(‘ ‘)

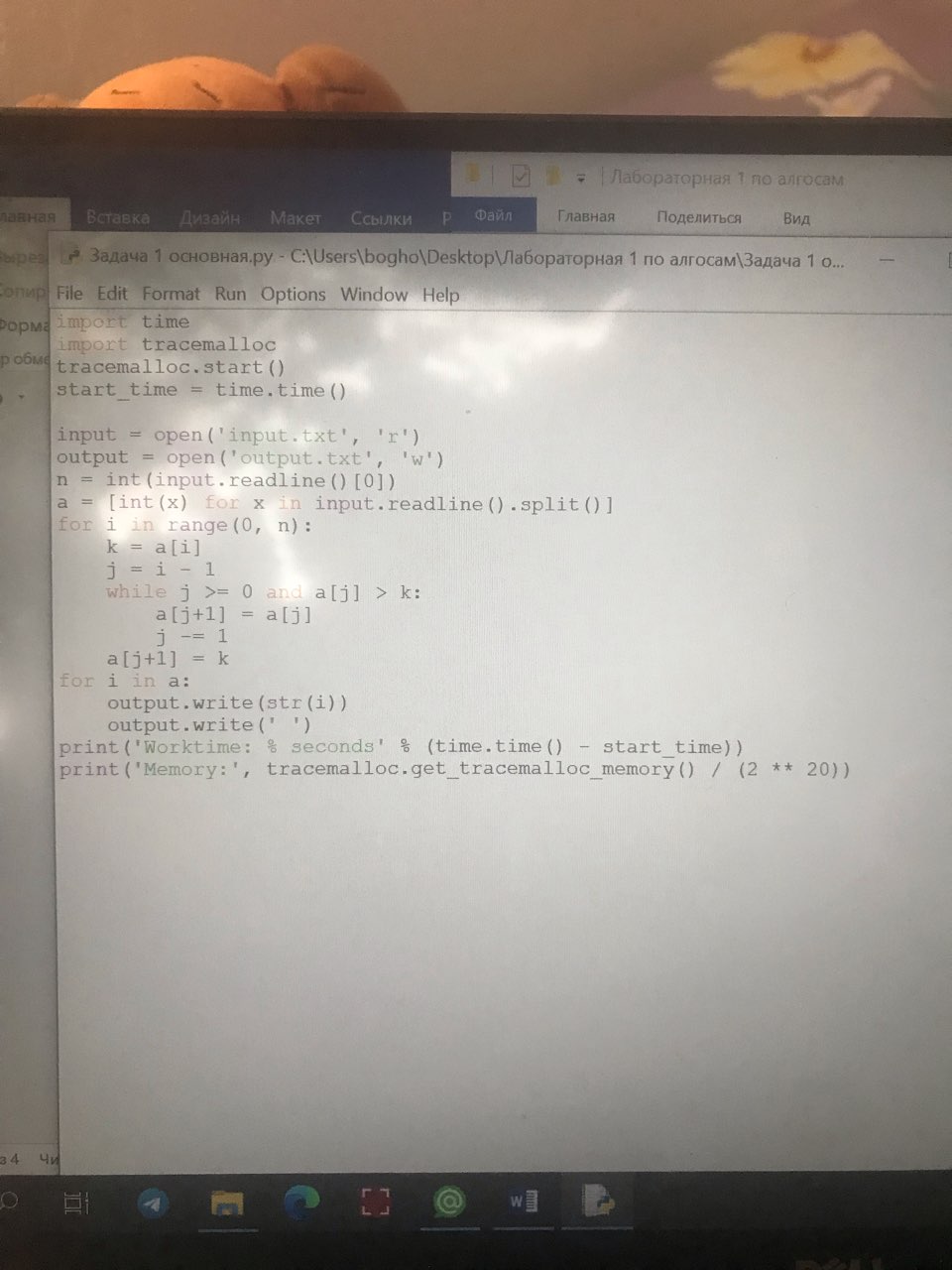
print(‘Worktime: % seconds’ % (time.time() – start\_time))

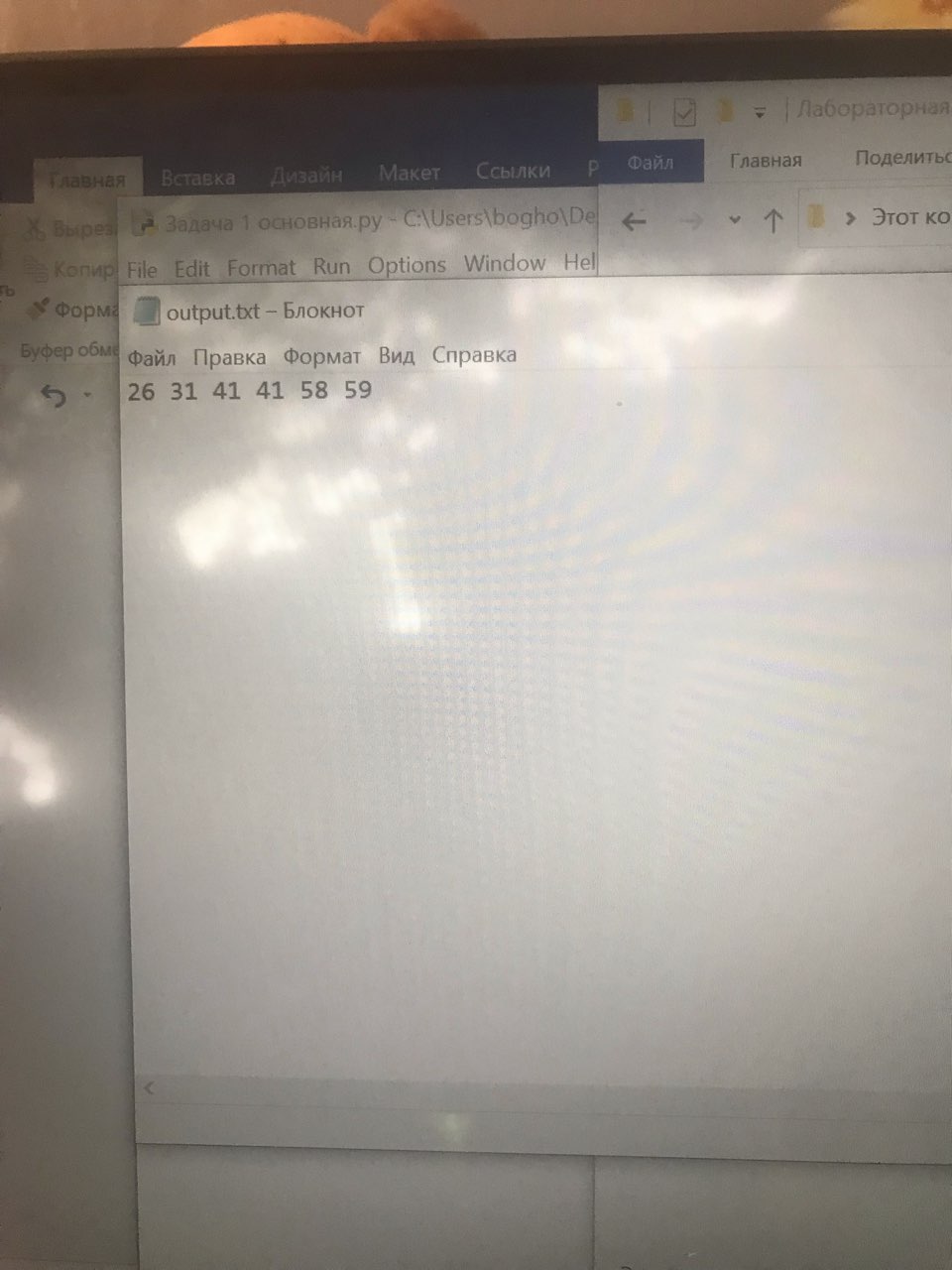
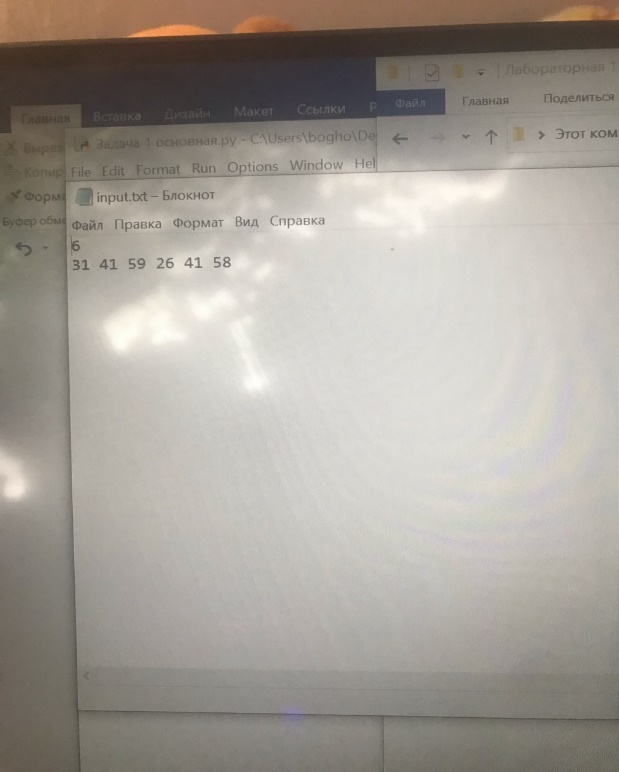
print(‘Memory:’, tracemalloc.get\_tracemalloc\_memory() / (2 \*\* 20))

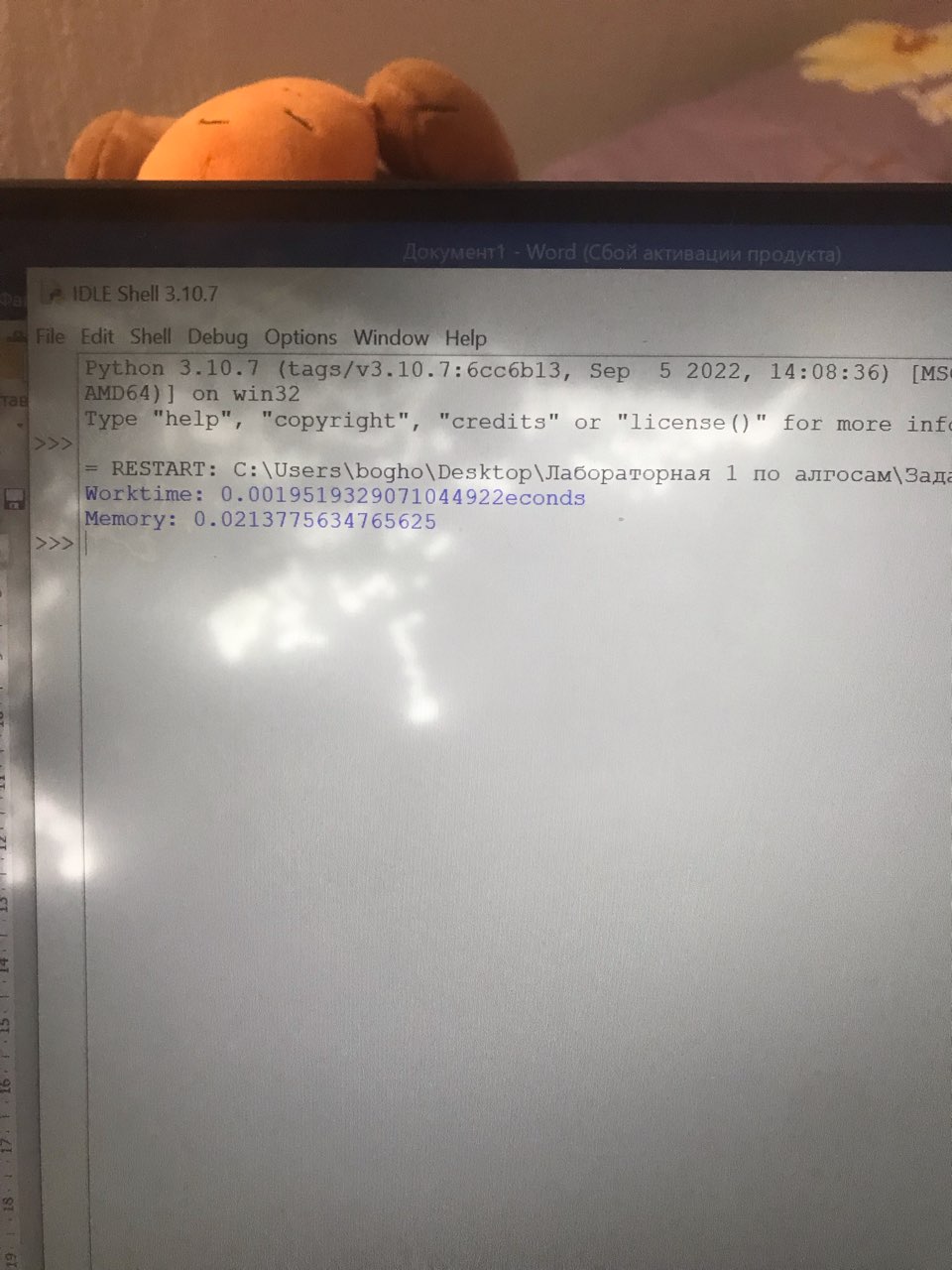
Текстовое обьяснение решения.

Открыла два файла, в которые будут вноситься данные. Прочитала строки в input.txt. Затем по условию метода сортировки вставкой отсортировала массив, так чтоб последующее число было больше предыдущего. Также высчитала время и память, затраченные на воспроизведение кода.

Input output файлы



****

****

Вывод по задаче: Я научилась делать сортировку вставкой.

**Задача 5. Сортировка выбором**

Рассмотрим сортировку элементов массива, которая выполняется следующим образом. Сначала определяется наименьший элемент массива, который ставится на место элемента A[1]. Затем производится поиск второго наименьшего элемента массива A, который ставится на место элемента A[2]. Этот процесс продолжается для первых n-1 элементов массива A.

Напишите код этого алгоритма, так же известного как сортировка выбором(selection sort). Определите время сортировки выбором в наихудшем случае и в среднем случае и сравните его со временем сортировки вставкой.

Формат входного и выходного файла и ограничения – как в задаче 1.

Листинг кода:

import tracemalloc

import time

start\_time = time.time()

tracemalloc.start()

input = open(‘input.txt’, ‘r’)

output = open(‘output.txt’, ‘w’)

n = int(input.readline() [0])

a = [int(x) for x in input.readline().split()]

for i in range(n – 1):

m = a[i]

p = i

for j in range(i+1, n):

if m > a[j]:

m = a[j]

p = j

if p != i:

t = a[i]

a[i] = a[p]

a[p] = t

for i in a:

output.write(str(i))

output.write(‘ ‘)

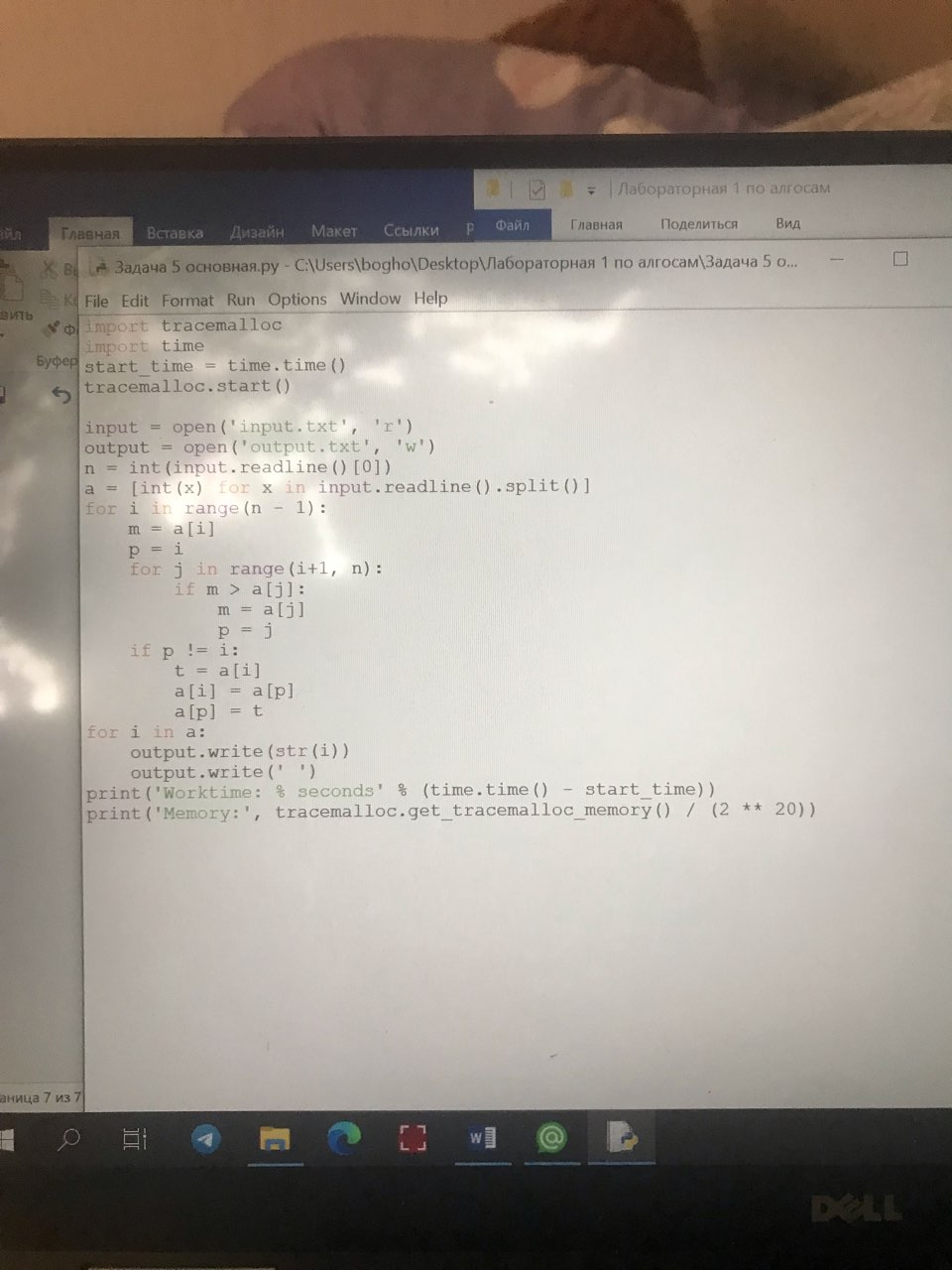
print(‘Worktime: % seconds’ % (time.time() – start\_time))

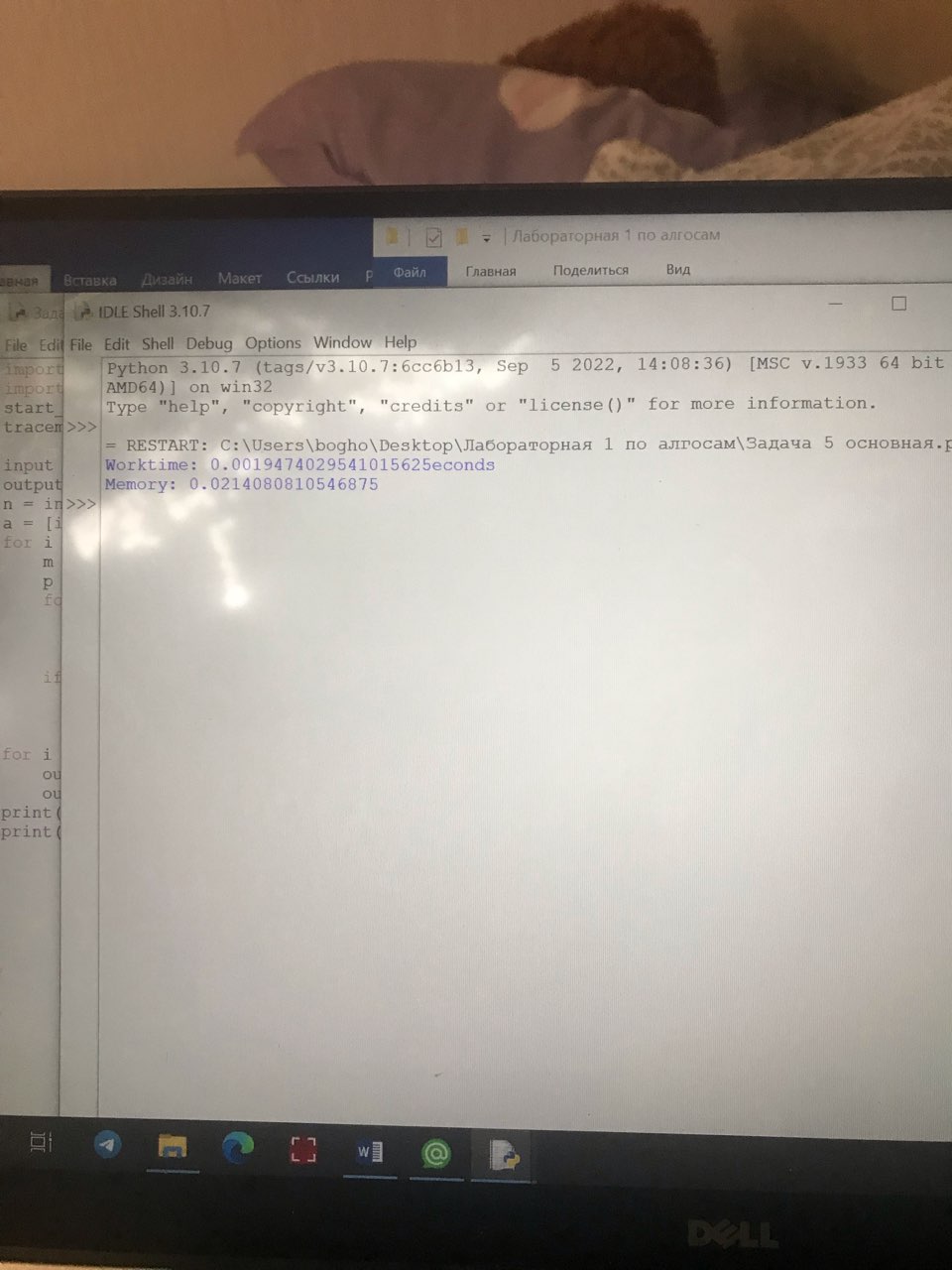
print(‘Memory:’, tracemalloc.get\_tracemalloc\_memory() / ( 2 \*\* 20))

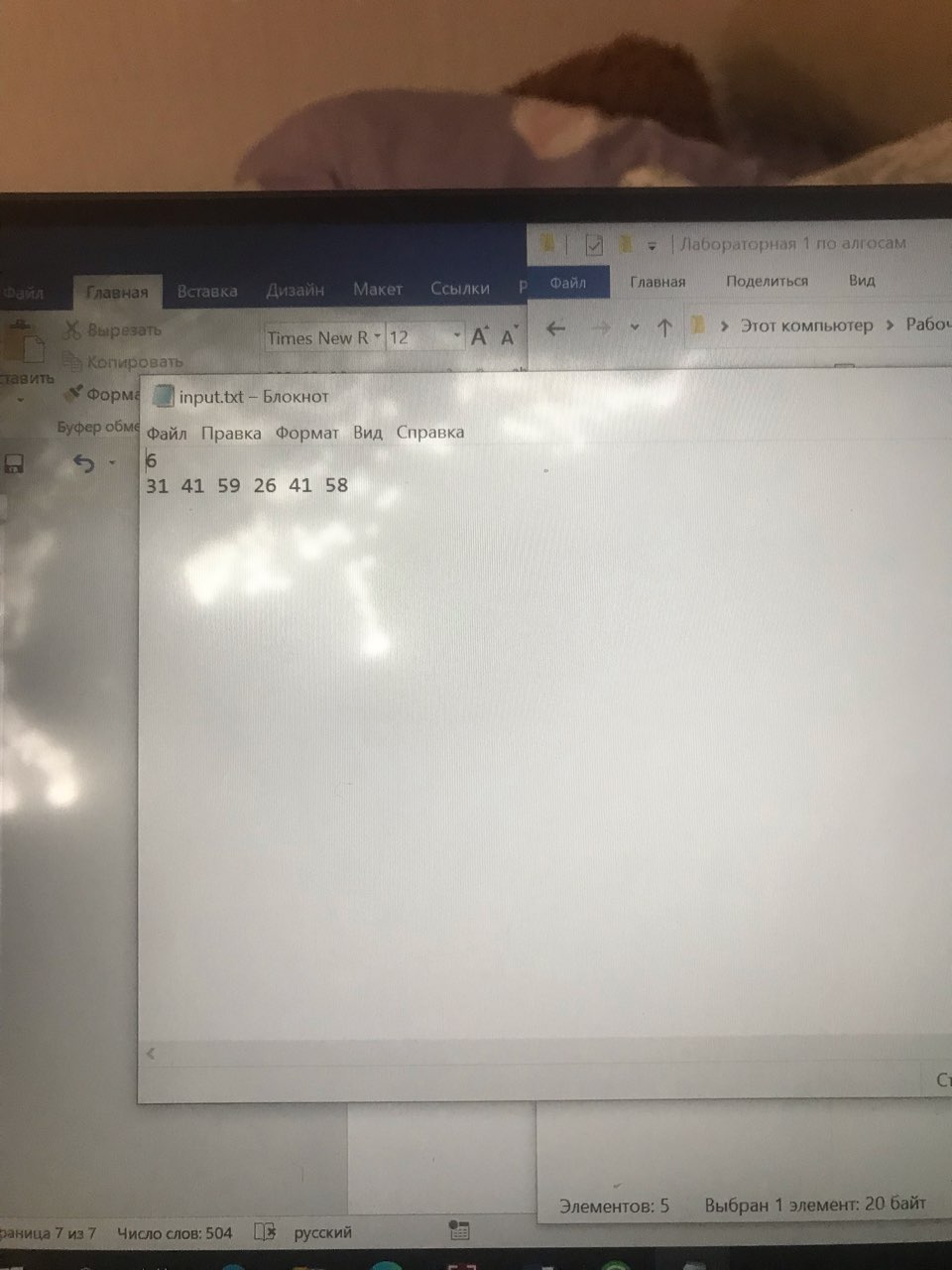
Текстовое обьяснение кода:

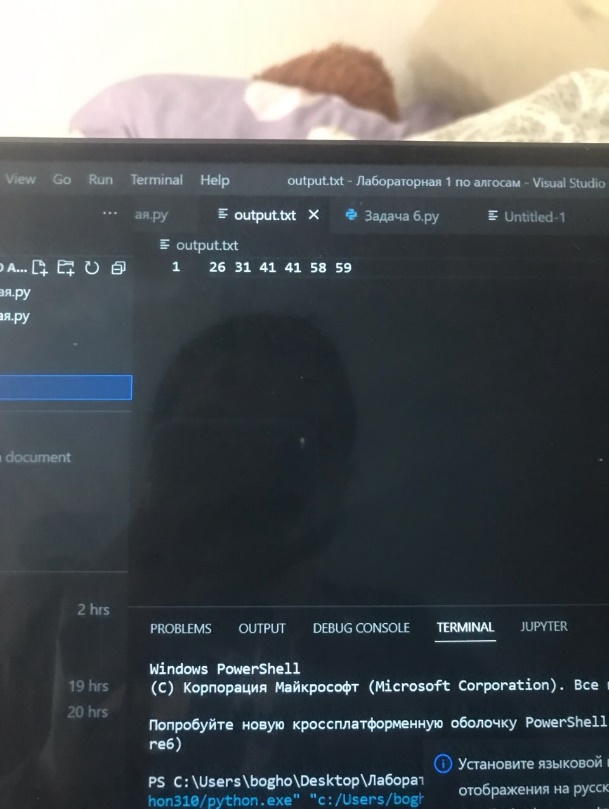
Сначала открыла два файла, в которые будут вноситься данные. Затем прочла их. Потом с помощью метода сортировки выбором, при котором последующий элемент выбирает из оставшихся элементов наименьший и перемещает его на позицию исходного числа. Также высчитала память и время, затраченные на воспроизведение кода.

Input output файлы:









Вывод: Я научилась производить сортировку выбором.

**Задача 6. Пузырьковая сортировка**

Пузырьковая сортировка представляет собой популярный, но не очень эффективный алгоритм сортировки. В его основе лежит многократная перестановка соседних элементов, нарушающих порядок сортировки. Вот псевдокод этой сортировки:

Напишите код на Python и докажите корректность пузырьковой сортировки. Для доказательства корректности процедуры вам необходимо доказать, что она завершается и что A’[1] <= A’[2]<= … <= A’[n], где A’ – выход процедуры Bubble\_Sort, a n – длина массива A.

Определите время пузырьковой сортировки в наихудшем случае и в среднем случае и сравните его со временем сортировки вставкой.

Формат входного и выходного файла и ограничения как в задаче 1.

Листинг:

import time

import tracemalloc

tracemalloc.start()

t\_start = time.perf\_counter()

with open(‘input.txt’) as f:

n = int(f.readline()[0])

A = [int(x) for x in f.readline().split(‘ ‘)]

def BubbleSort(A):

for i in range(n):

for j in range(0, n – i – 1):

if A[j] > A[j+1]:

b = A[j]

A[j] = A[j+1]

A[j+1] = b

BubbleSort(A)

out = open(‘output.txt’, ‘w’)

for j in A:

out.write(str(j))

out.write(‘ ‘)

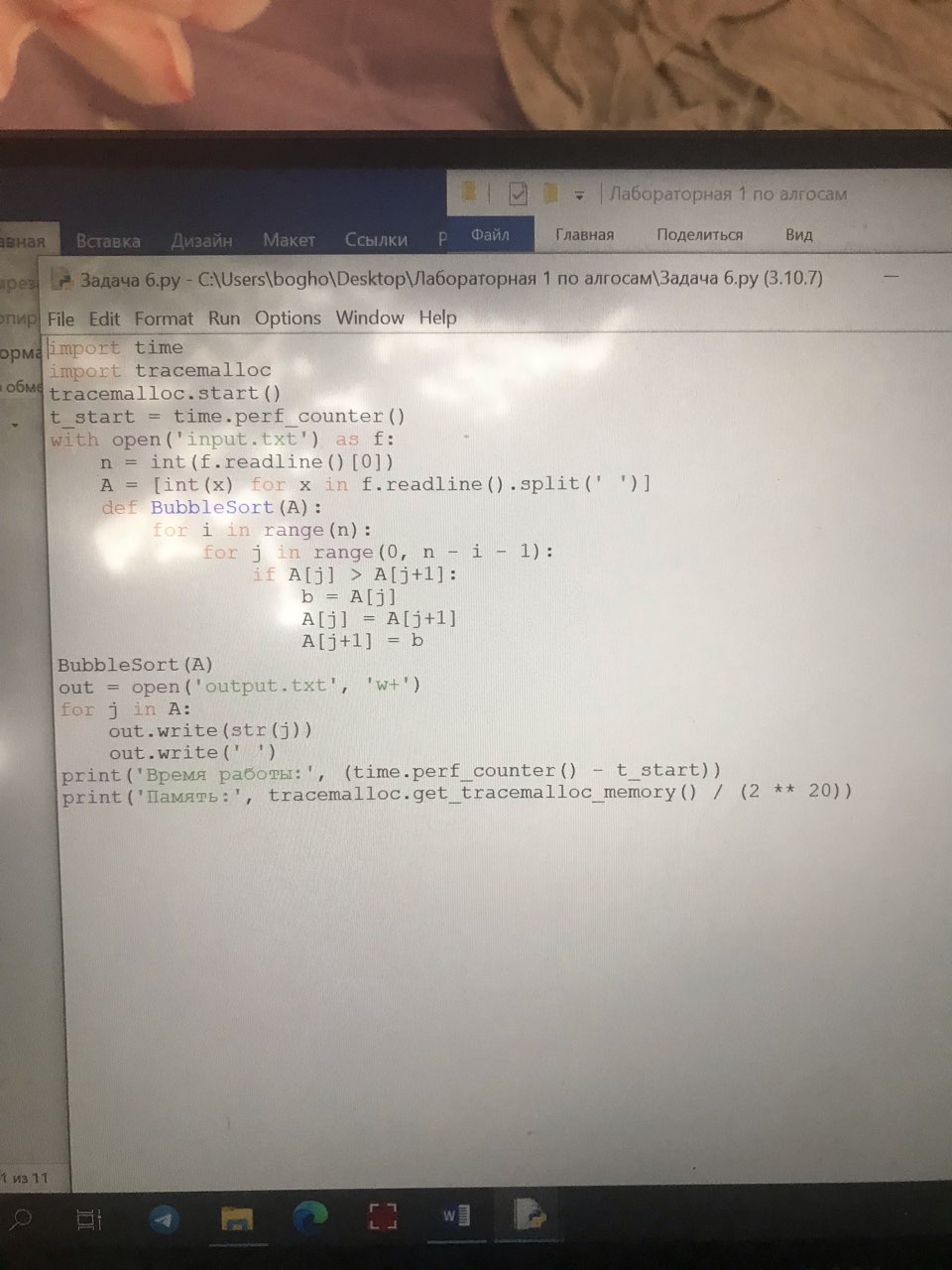
print(‘Время работы:’, (time.perf\_counter() – t\_start))

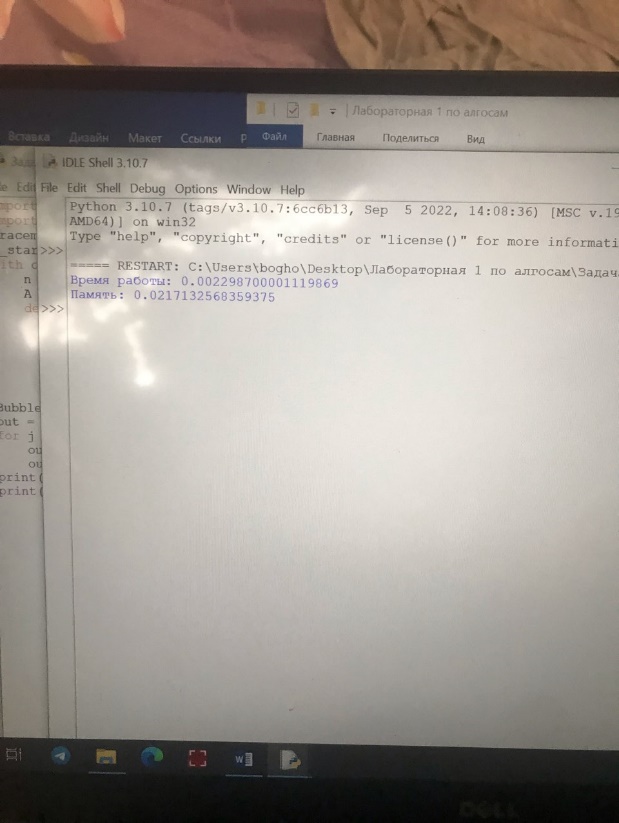
print(‘Память:’, tracemalloc.get\_tracemalloc\_memory() / ( 2 \*\* 20))

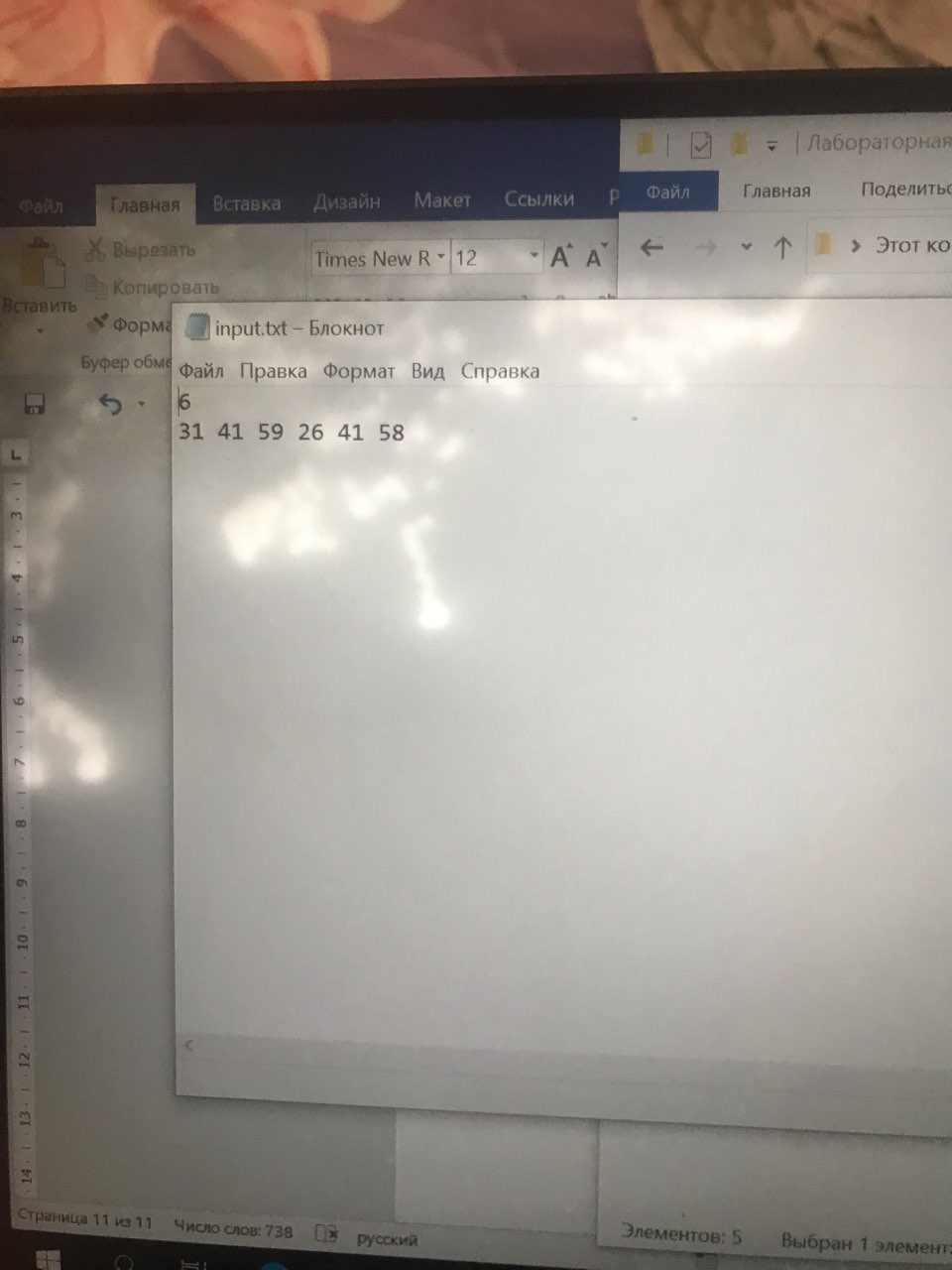
Текстовое обьяснение решения:

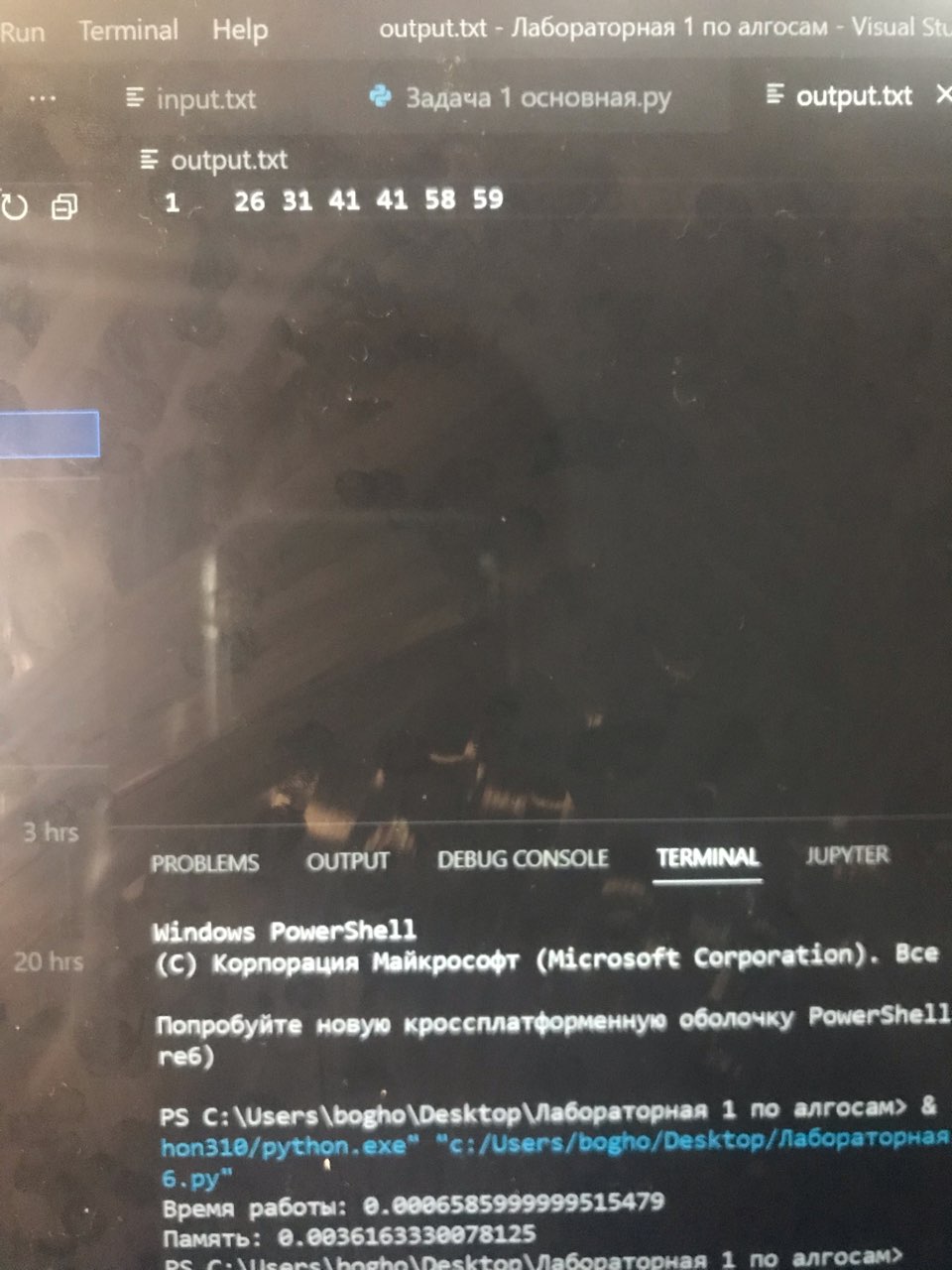
Ввела инструмент отладки для отслеживания блоков памяти. Открыла два файла, в которые будут вноситься данные. Прочла их. Затем с помощью метода пузырьковой сортировки, которая позволяет попарно сравнить элементы, менять их местами в случае необходимости и устанавливать их в верной последовательности, решила задачу. В конце также посчитала время и память, затраченные на выполнение кода.

Input output файлы









Вывод по задаче: Я научилась работать с методом пузырьковой сортировки.

Общий вывод: Я научилась работать с методами сортивки вставкой, выбором и пузырьковой сортировкой.