**Практическая работа №5**

**(Парфенов А.К. У-213)**

№1

1. По данному целому числу N распечатайте все квадраты натуральных чисел, не превосходящие N, в порядке возрастания.

# -\*- coding: utf-8 -\*-

def d1():

    i = 1

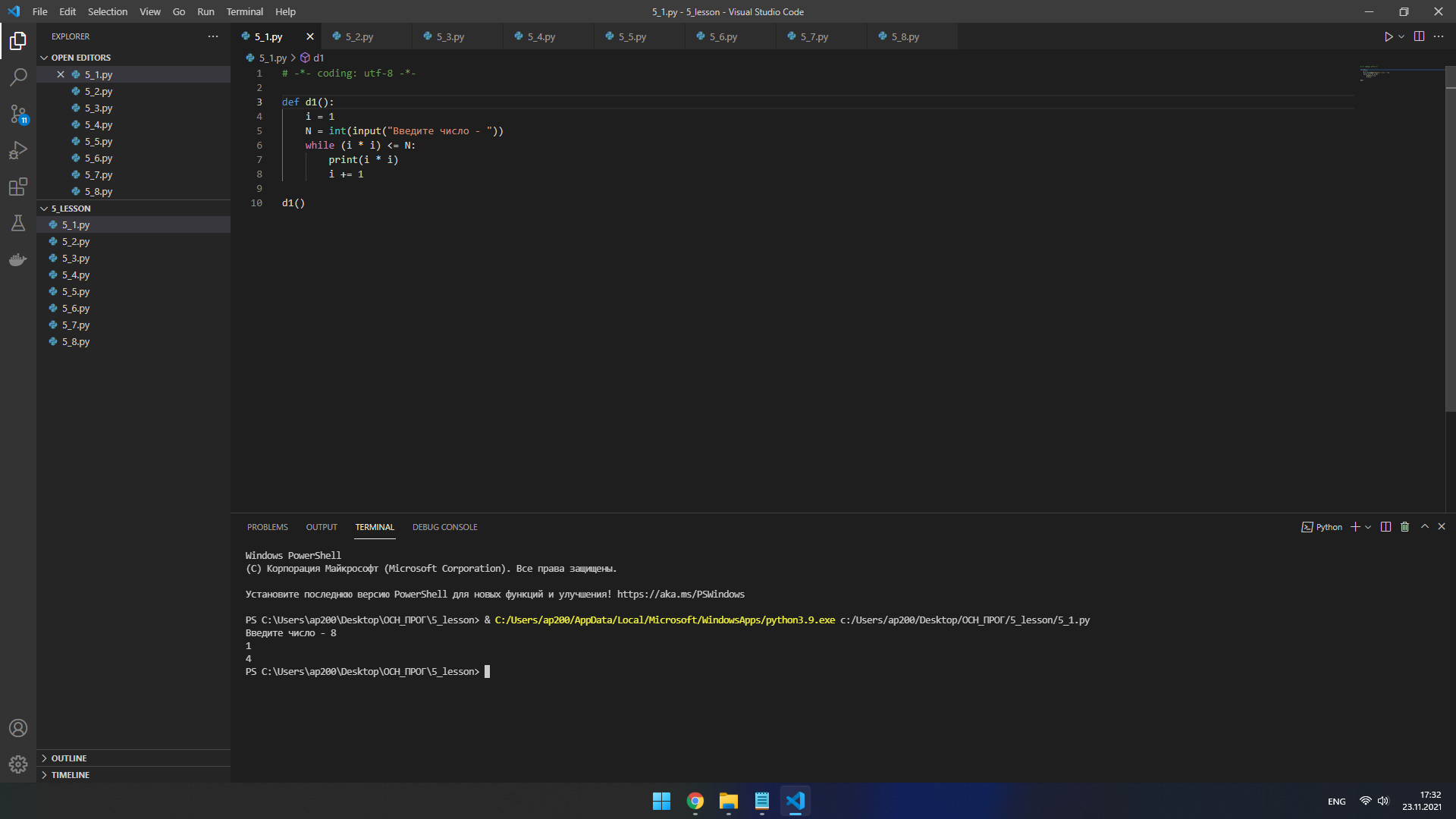
    N = int(input("Введите число - "))

    while (i \* i) <= N:

        print(i \* i)

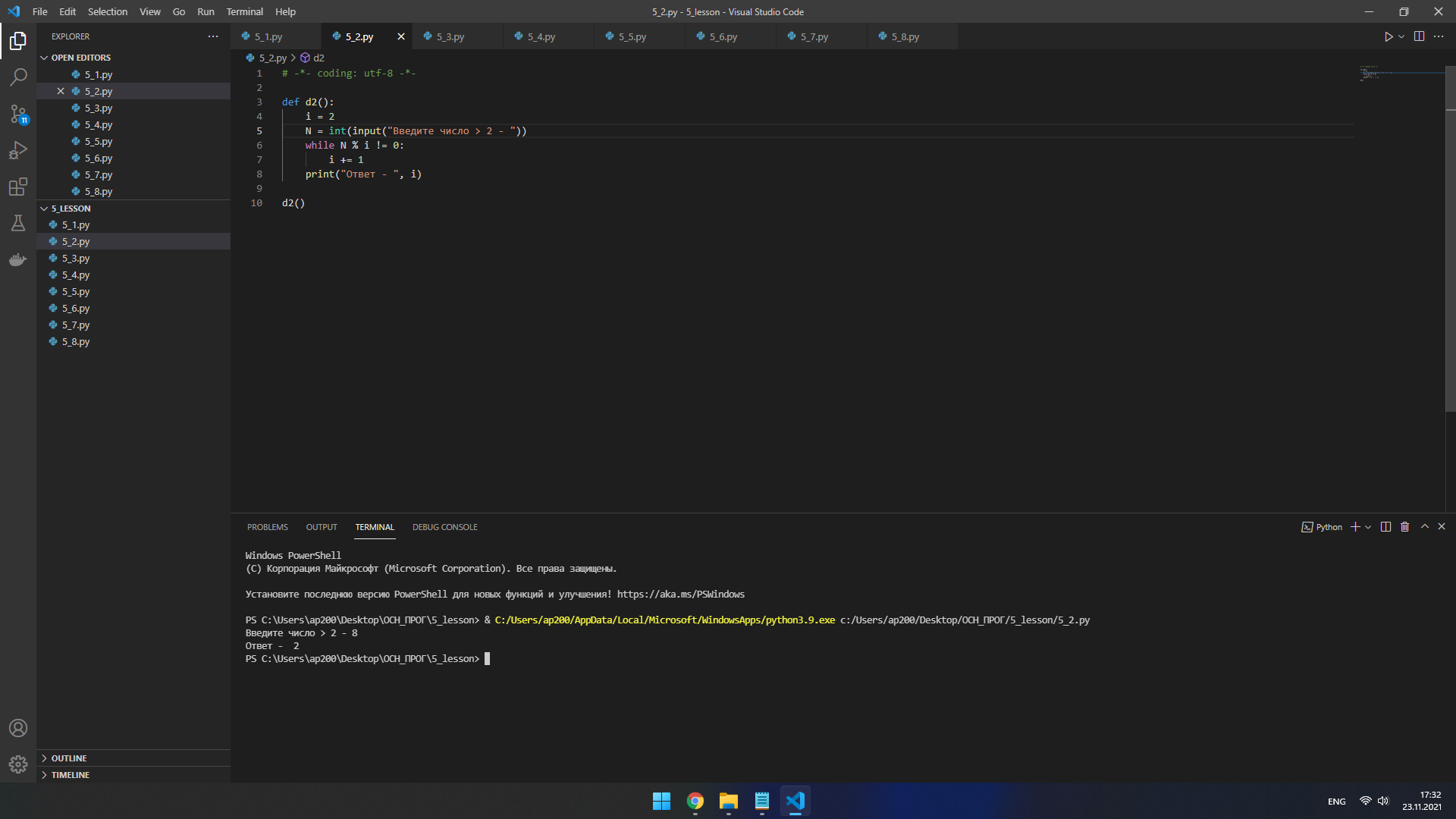
        i += 1

d1()



№2

1. Дано целое число, не меньшее 2. Выведите его наименьший натуральный делитель, отличный от 1.
2. # -\*- coding: utf-8 -\*-
3. def d2():
4. i = 2
5. N = int(input("Введите число > 2 - "))
6. while N % i != 0:
7. i += 1
8. print("Ответ - ", i)
9. d2()



№3

1. По данному натуральному числу N найдите наибольшую целую степень двойки, не превосходящую N. Выведите показатель степени и саму степень. Операцией возведения в степень пользоваться нельзя!

# -\*- coding: utf-8 -\*-

def d3():

    N = int(input("Введите число - "))

    alll = 2

    step = 1

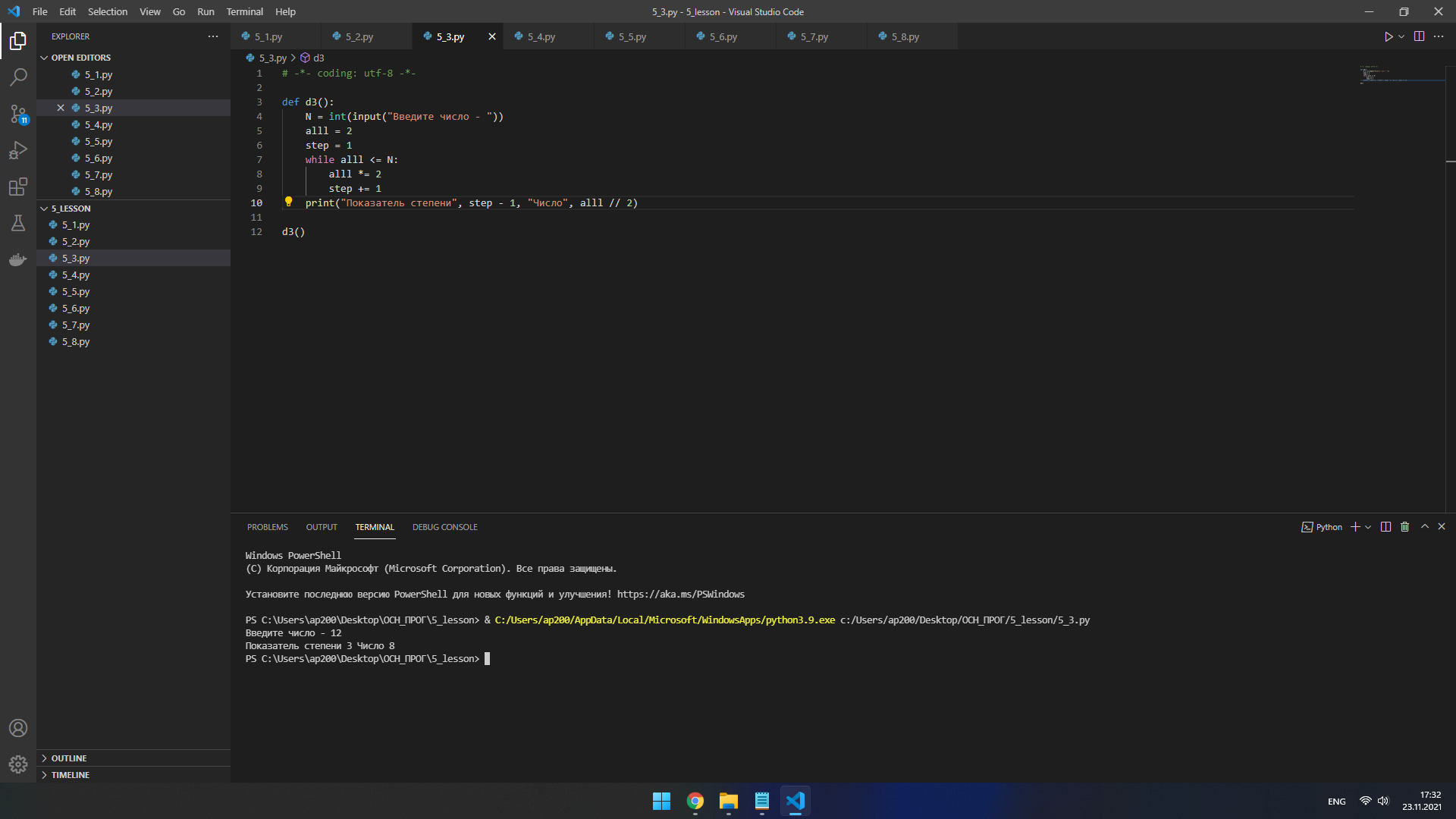
    while alll <= N:

        alll \*= 2

        step += 1

    print("Показатель степени", step - 1, "Число", alll // 2)

d3()



№4

1. В первый день спортсмен пробежал *x* километров, а затем он каждый день увеличивал пробег на 10% от предыдущего значения. По данному числу *y* определите номер дня, на который пробег спортсмена составит не менее *y* километров. Программа получает на вход действительные числа *x* и *y* и должна вывести одно натуральное число.

# -\*- coding: utf-8 -\*-

def d4():

    x = int(input("Введите первое число - "))

    y = int(input("Введите второе число - "))

    numb = 1

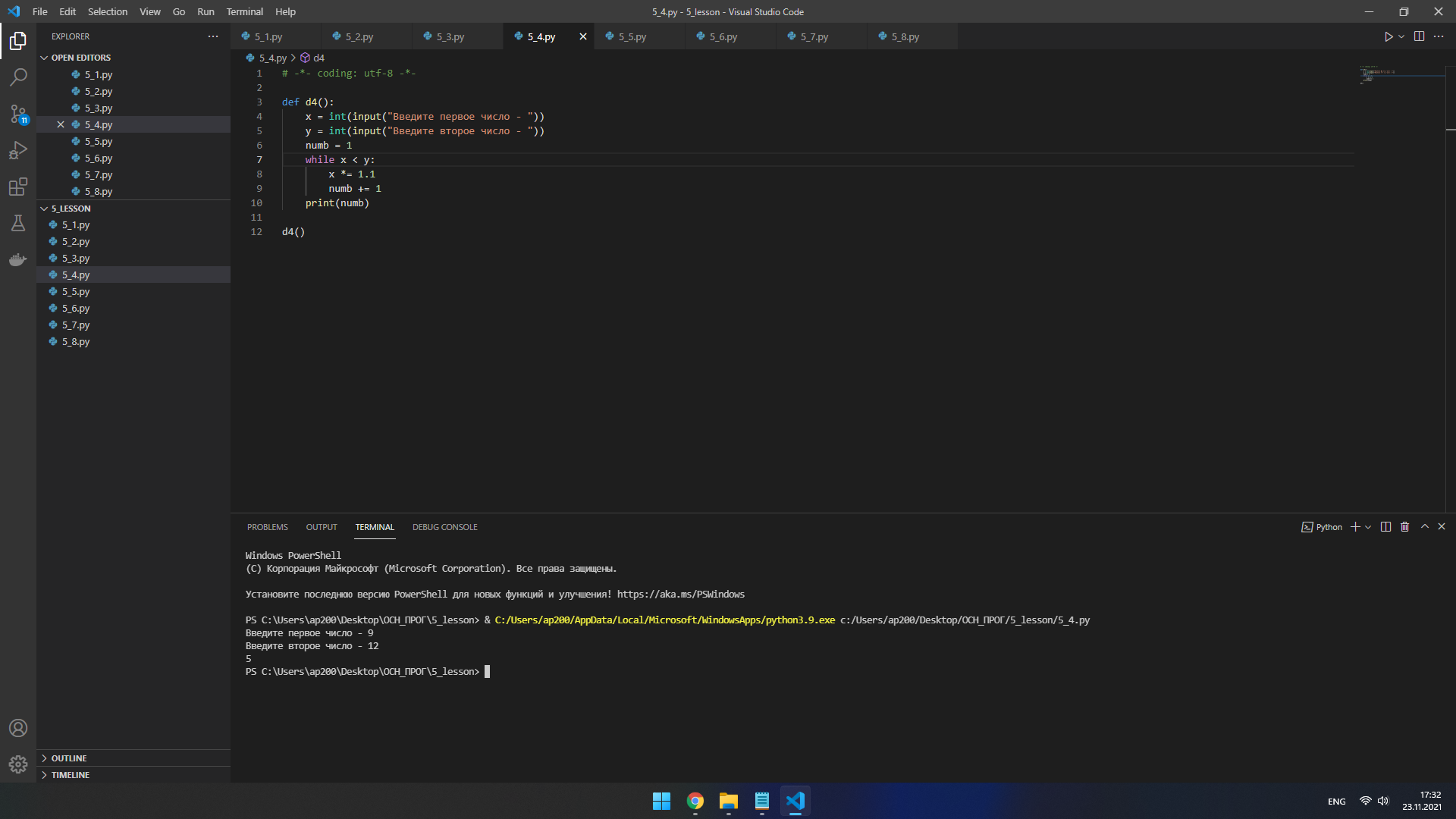
    while x < y:

        x \*= 1.1

        numb += 1

    print(numb)

d4()



№5

1. Программа получает на вход последовательность целых неотрицательных чисел, каждое число записано в отдельной строке. Последовательность завершается числом 0, при считывании которого программа должна закончить свою работу и вывести количество членов последовательности (не считая завершающего числа 0). Числа, следующие за числом 0, считывать не нужно.

# -\*- coding: utf-8 -\*-

def d5():

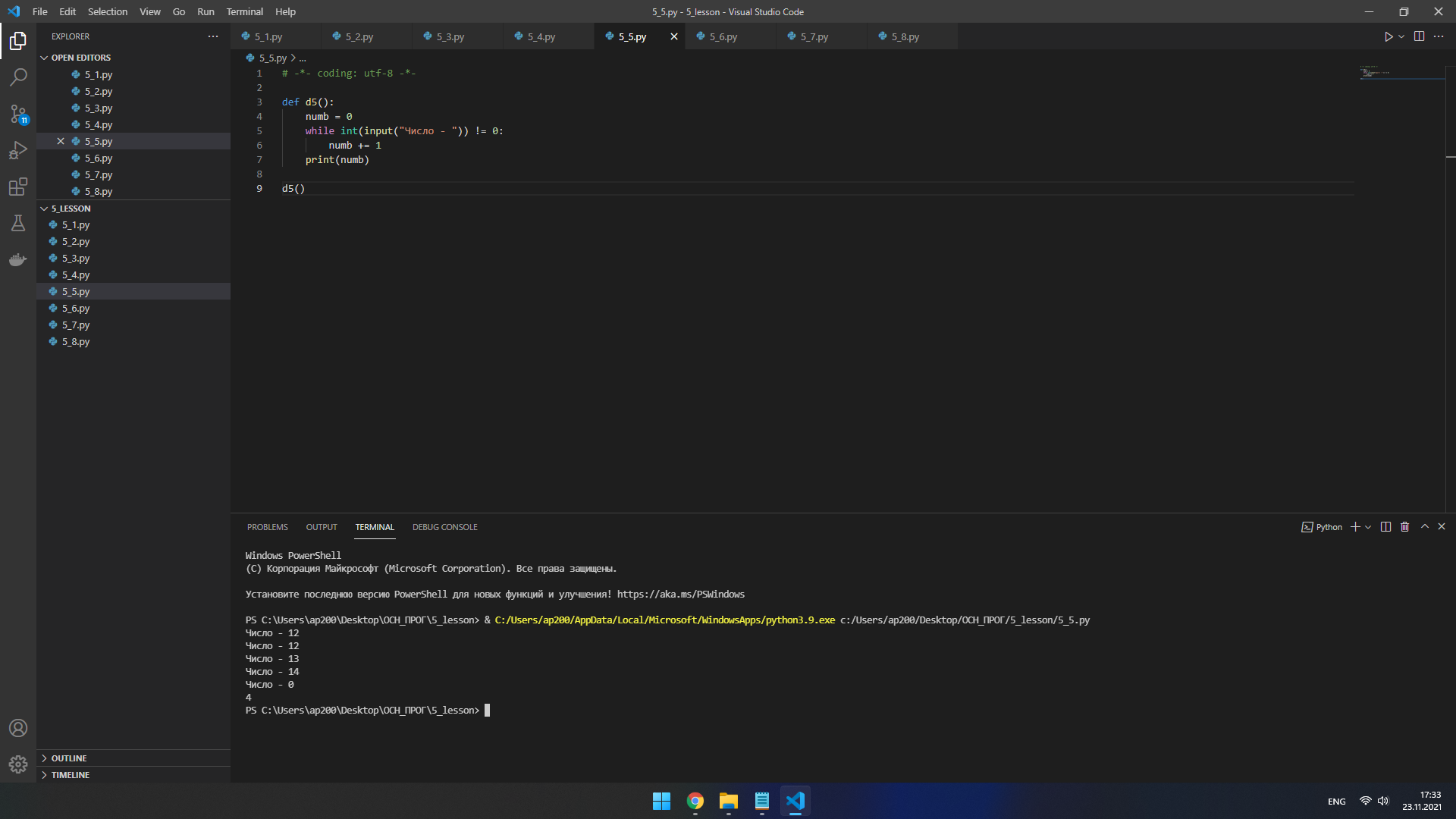
    numb = 0

    while int(input("Число - ")) != 0:

        numb += 1

    print(numb)

d5()



№6

1. Определите среднее значение всех элементов последовательности, завершающейся числом 0.

# -\*- coding: utf-8 -\*-

def d6():

    k\_num = 0

    summ = 0

    while True:

        x = int(input("Число - "))

        if x == 0:

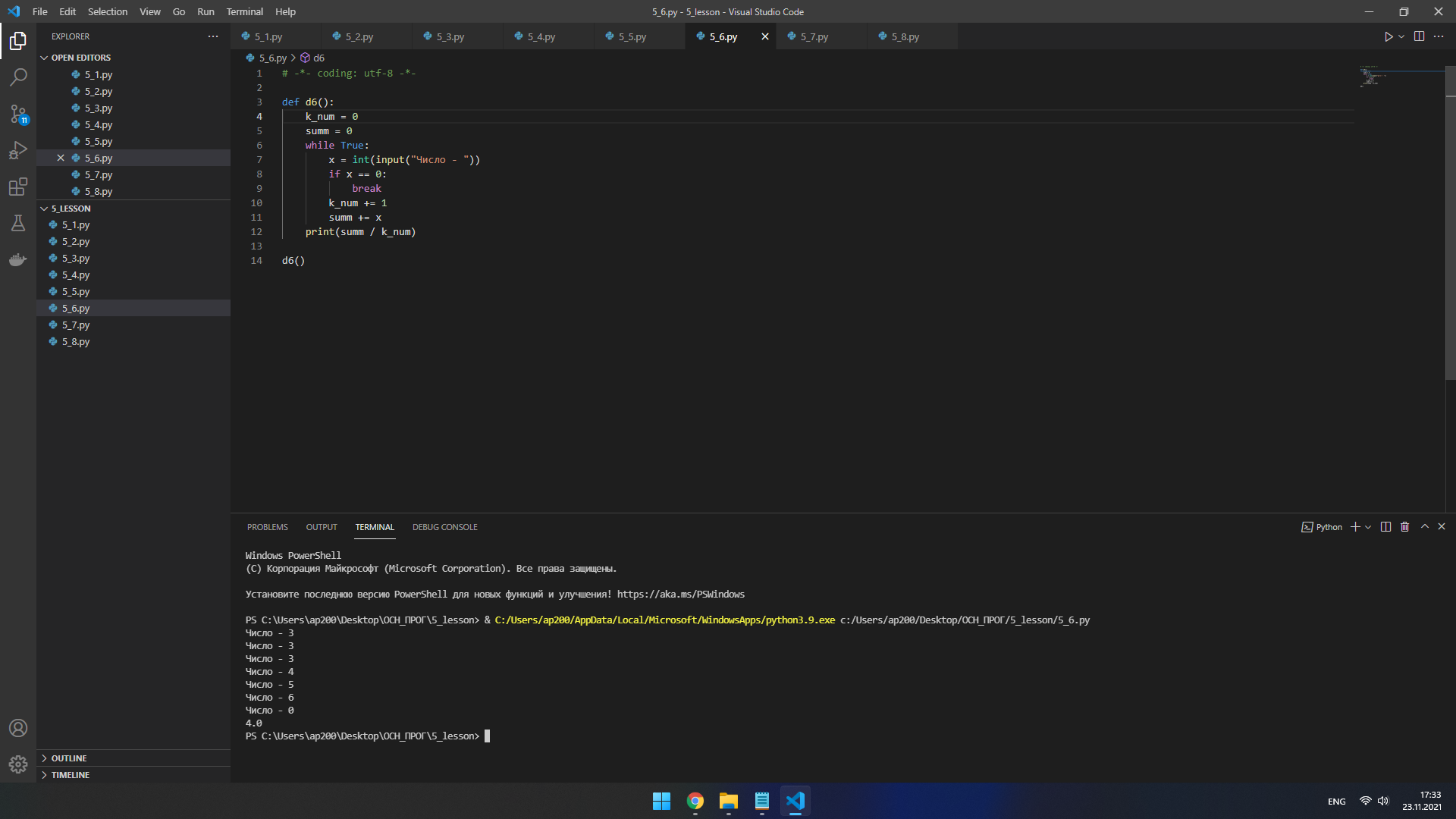
            break

        k\_num += 1

        summ += x

    print(summ / k\_num)

d6()



№7

1. Последовательность состоит из натуральных чисел и завершается числом 0. Определите, сколько элементов этой последовательности больше предыдущего элемента.

# -\*- coding: utf-8 -\*-

def d7():

    numb = 0

    loop\_first = True

    while True:

        x = int(input("Число - "))

        if x == 0:

            break

        if(loop\_first != True):

            if(x > container):

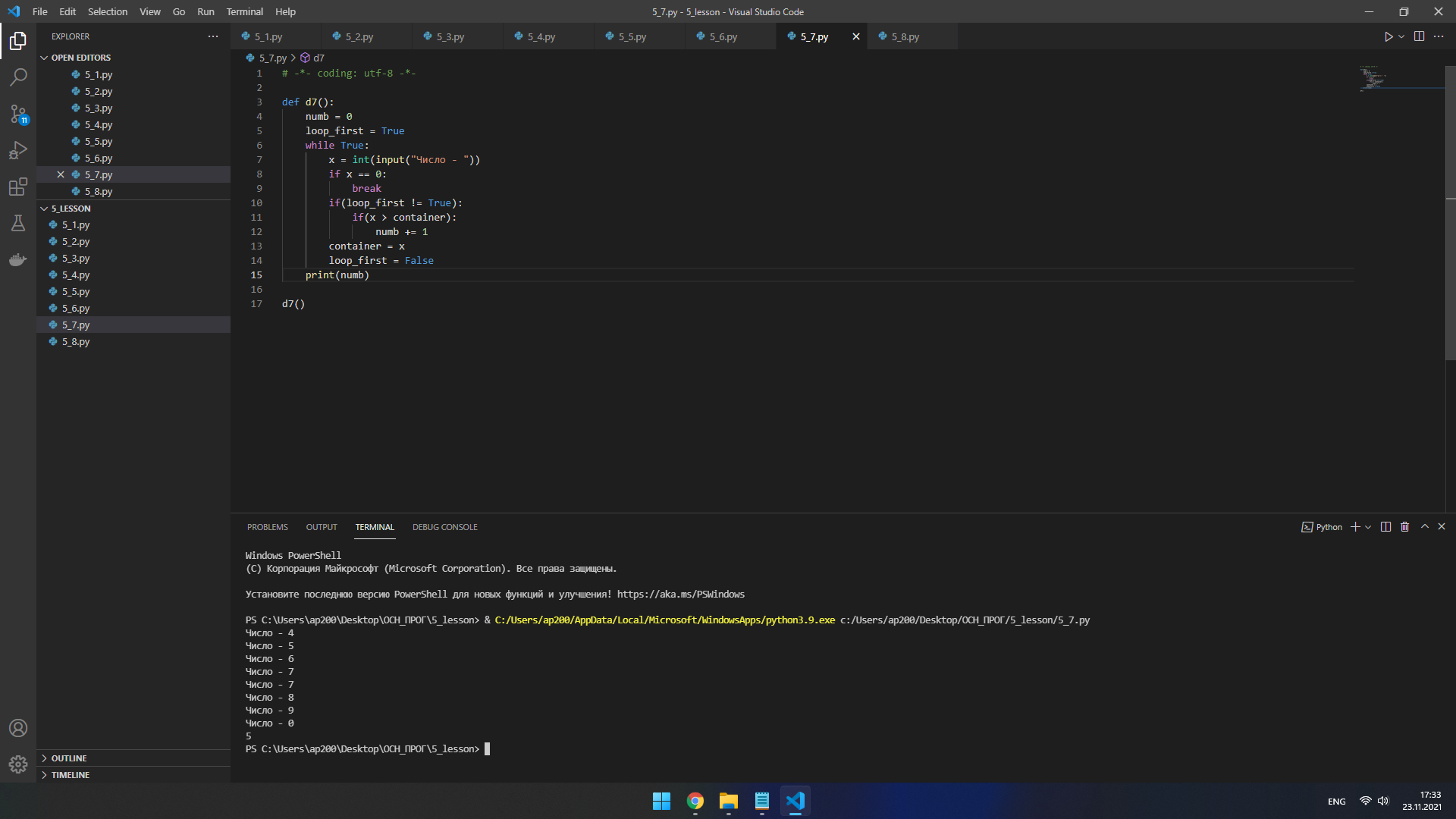
                numb += 1

        container = x

        loop\_first = False

    print(numb)

d7()



№8

1. Дана последовательность натуральных чисел, завершающаяся числом 0. Определите, какое наибольшее число подряд идущих элементов этой последовательности равны друг другу.
2. # -\*- coding: utf-8 -\*-
3. def d8():
4. numb = 1
5. maxStreak = 0
6. loop\_first = True
7. while True:
8. x = int(input("Введите число - "))
9. if x == 0:
10. break
11. if(loop\_first != True):
12. if(x == container):
13. numb += 1
14. else:
15. if(maxStreak < numb):
16. maxStreak = numb
17. numb = 1
18. container = x
19. loop\_first = False
20. print(max(maxStreak, numb))
21. d8()

