Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №7**

**«ИЗУЧЕНИЕ ФРАКТАЛОВ»**

**ПО МДК 05.02 РАЗРАБОТКА КОДА ИНФОРМАЦИОННЫХ СИСТЕМ**

Выполнил: студент учебной группы

ИСПк-205-52-00

Ляпустин Илья Максимович

Преподаватель:

Сергеева Елизавета Григорьевна

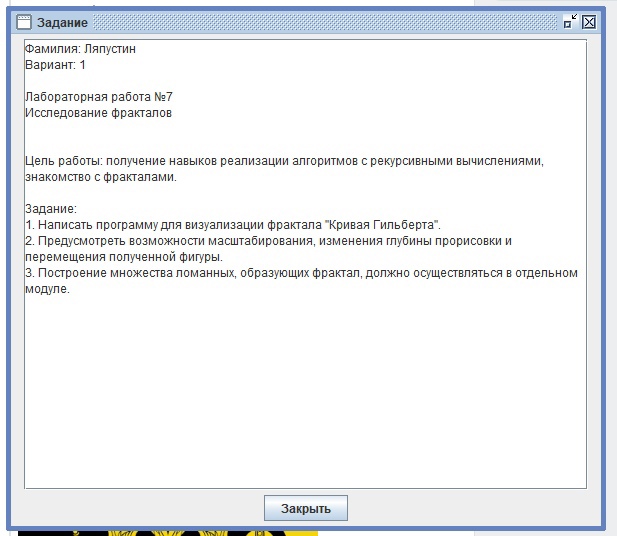
Киров

2023

**1.ЦЕЛЬ ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЫ**

Цель работы: получение навыков реализации алгоритмов с рекурсивными вычислениями, знакомство с фракталами.

**2.ФОРМУЛИРОВКА ЗАДАНИЯ**



**3.КОД ПРОГРАММЫ**

import turtle

class LSystem2D:

def \_\_init\_\_(self, t, axiom, width, length, angle):

self.axiom = axiom

self.state = axiom

self.width = width

self.length = length

self.angle = angle

self.t = t

self.rules = {}

self.t.pensize(self.width)

def add\_rules(self, \*rules):

for key, value in rules:

self.rules[key] = value

def generate\_path(self, n\_iter):

for n in range(n\_iter):

for key, value in self.rules.items():

self.state = self.state.replace(key, value.lower())

self.state = self.state.upper()

def draw\_turtle(self, start\_pos, start\_angle):

turtle.tracer(1, 0)

self.t.up()

self.t.setpos(start\_pos)

self.t.seth(start\_angle)

self.t.down()

for move in self.state:

if move == 'F':

self.t.forward(self.length)

elif move == 'S':

self.t.up()

self.t.forward(self.length)

self.t.down()

elif move == '+':

self.t.left(self.angle)

elif move == '-':

self.t.right(self.angle)

turtle.done()

width = int(input())

height = int(input())

screen = turtle.Screen()

screen.setup(width, height, 0, 0)

t = turtle.Turtle()

t.ht()

pen\_width = 2

f\_len = 5

angle = 90

axiom = "X"

l\_sys = LSystem2D(t, axiom, pen\_width, f\_len, angle)

l\_sys.add\_rules(("X", "-YF+XFX+FY-"), ("Y", "+XF-YFY-FX+"))

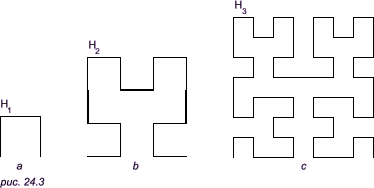
l\_sys.generate\_path(int(input()))

print(l\_sys.state)

l\_sys.draw\_turtle( (0, 0), -180)

**4.ОПИСАНИЕ РАБОТЫ РЕКУРСИИ**

Рекурсия может быть использована для получения линейного рисунка, известного под названием *кривая Гильберта*. Кривая Гильберта основана на изображении буквы П, вычерченной в виде трех сторон квадрата, как показано на рисунке ниже. Существуют кривые Гильберта порядков 1, 2, ..., обозначаемые как H1, H2,... На **рис. 24.3** изображена кривая порядка H2, в которой некоторые отрезки прямых линий вычерчены в виде толстых линий. Это так называемые *связки* (в действительности связки должны иметь одинаковую толщину с другими отрезками, здесь же они показаны утолщенными единственно с целью демонстрации способа получения H2 из H1).



Видно, что H2 можно рассматривать как большую букву П, четыре части которой заменены меньшими по размеру буквами П. Эти меньшие буквы П соединены тремя связками. Каждая сторона меньшей буквы П имеет ту же длину, что и связка, они в три раза меньше стороны квадрата, в который вписывается H2. Применим ту же процедуру к каждой из четырех букв П, составляющих H2, то есть каждую букву П в H2 заменим меньшей H2, одновременно уменьшим длину связок так, чтобы их длины стали равными длине элементарного отрезка прямой линии, которые содержатся в трех малых фигурах H2. Таким образом мы получим фигуру H3, показанную на рисунке.

Теперь все элементарные отрезки в семь раз меньше, чем длина сторон квадрата, в который вписывается фигура H3. Отсюда получаем, что коэффициенты уменьшения для этих элементарных отрезков в фигурах H1, H2, H3, ... образуют ряд чисел: 1, 3, 7, ..., то есть в общем случае коэффициент уменьшения для фигуры Hn может быть вычислен по формуле: 2n – 1.

**5.РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ ПРОГРАММЫ**

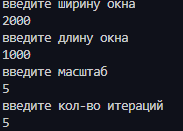


Рисунок 6 – ввод данных в программу (1)

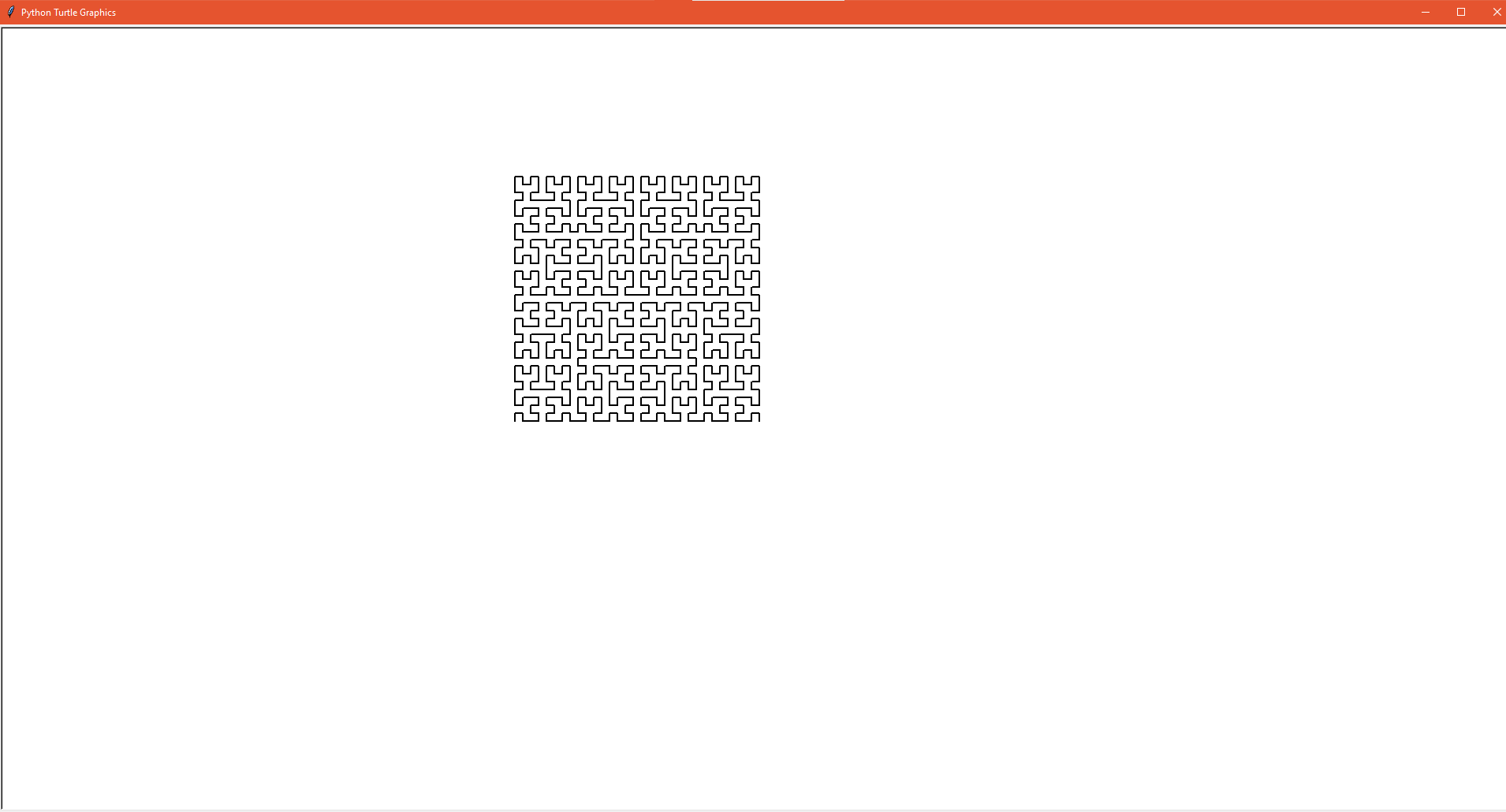


Рисунок 7 – вывод результата программой (1)

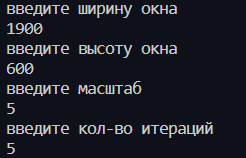


Рисунок 8 – ввод данных в программу (2)

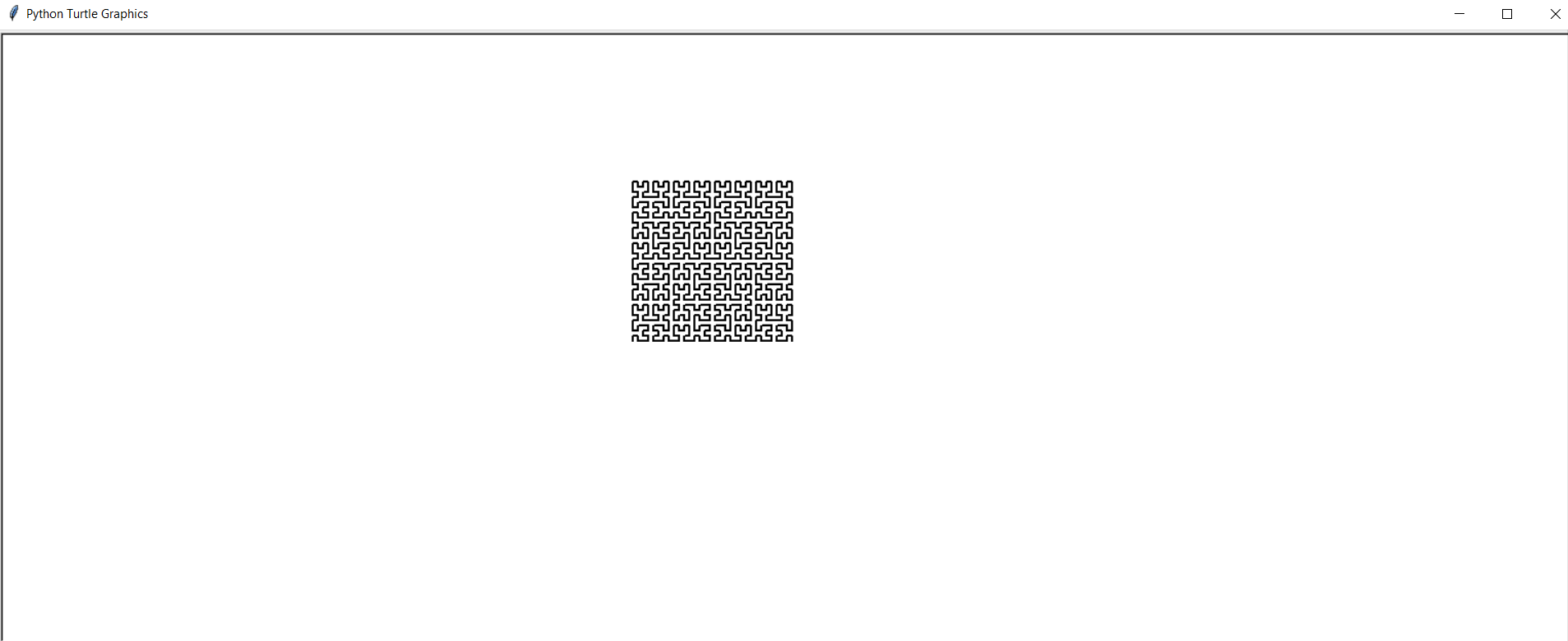


Рисунок 9 – вывод результата программой (2)

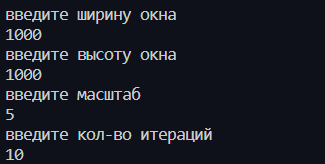


Рисунок 10 – ввод данных в программу (3)

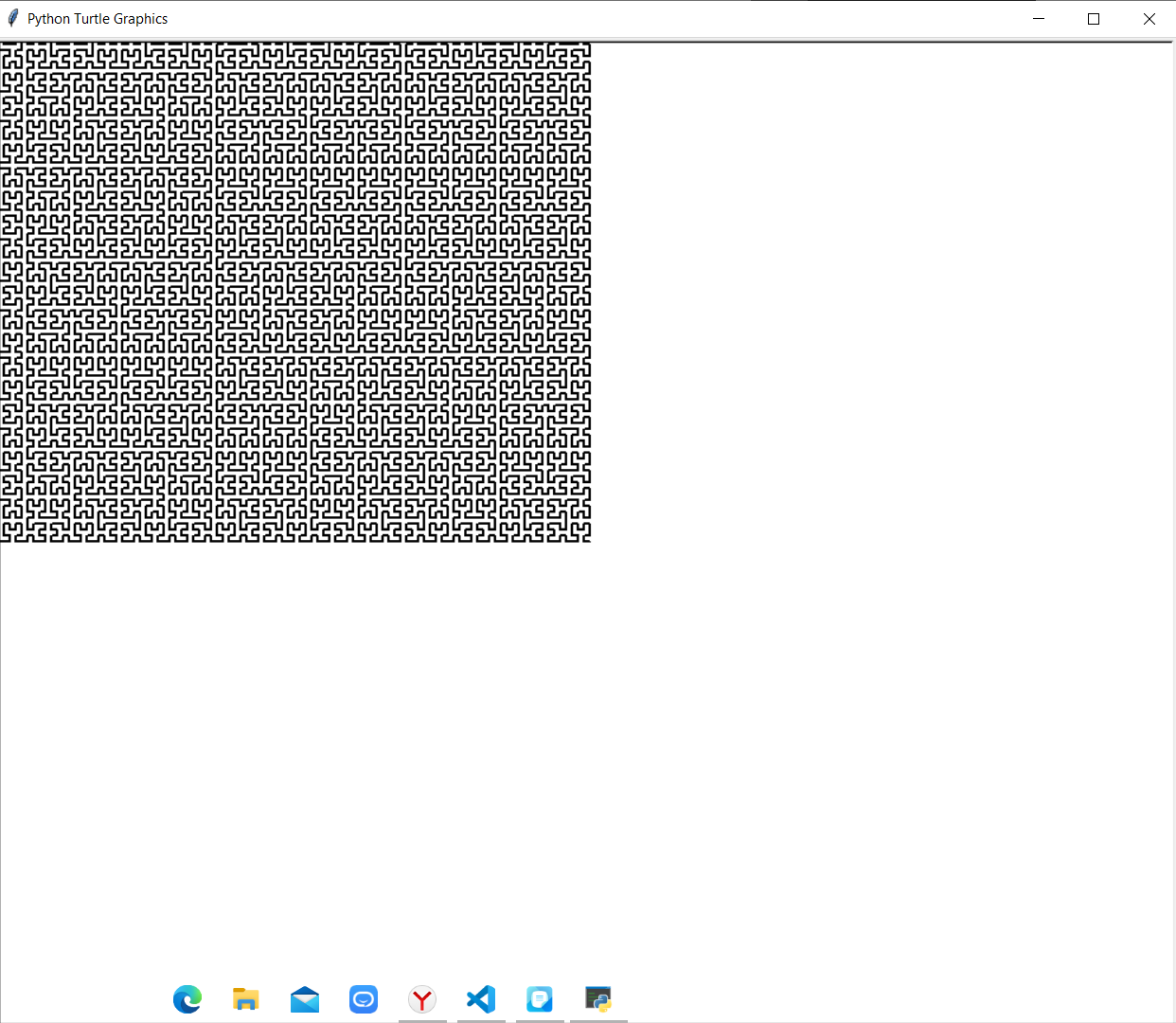


Рисунок 11 – вывод результата программой (3)

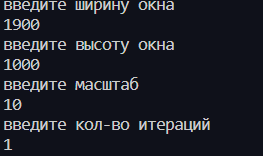


Рисунок 10 – ввод данных в программу (4)

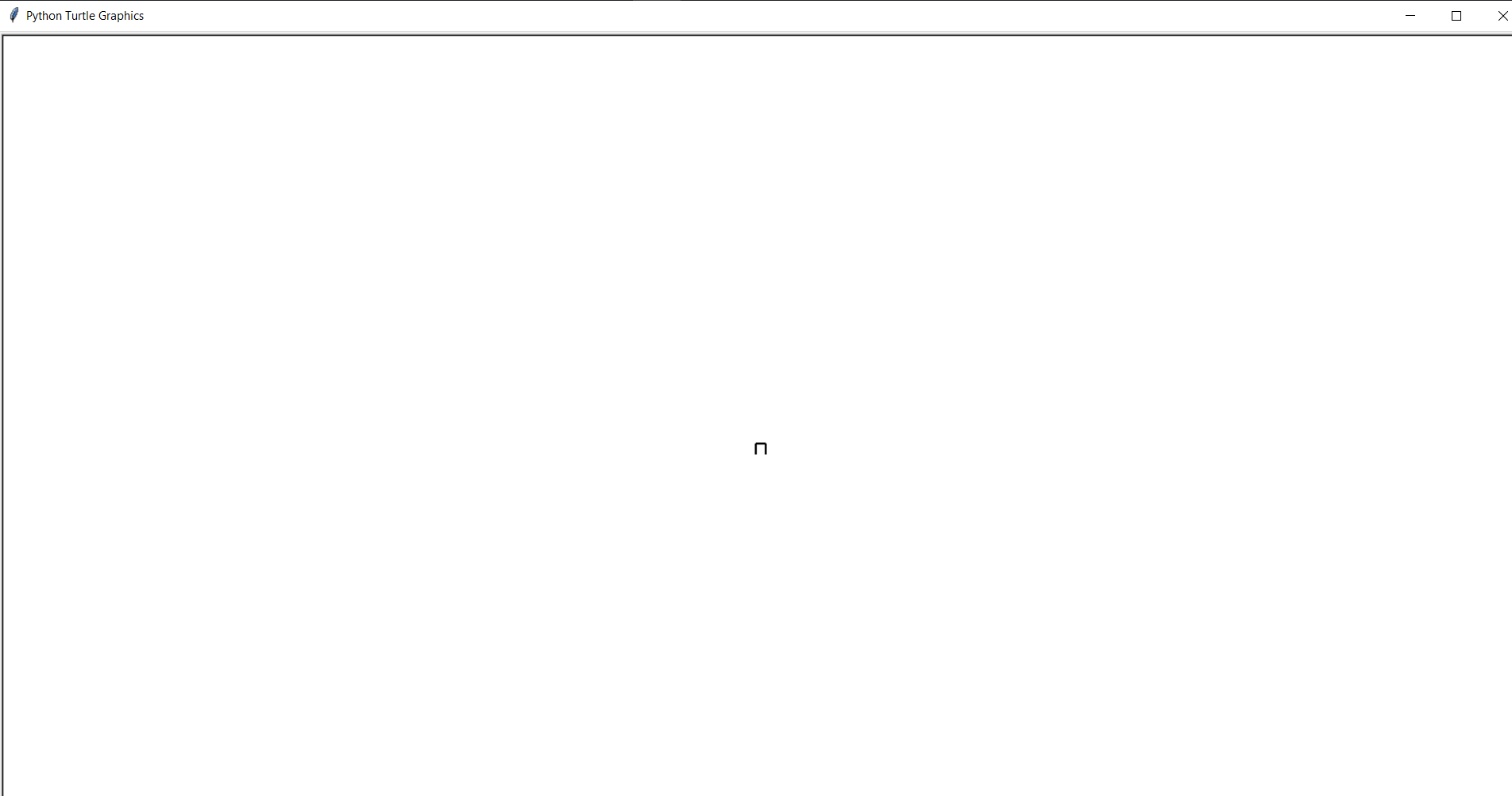


Рисунок 12 – вывод результата программой (4)

**6. ВЫВОД**

На данной домашней контрольной работе был освоен синтаксис построения рекурсий, изучены способы передачи данных в программы, получены навыки организации минимального пользовательского интерфейса языка программирования Python. Была создана программа, рисующая фрактал «кривая Гильберта». Параметры выполнения должны вводиться пользователем.

Была использована библиотека «Turtle» для визуализации результата.

После того как мы создали программу и схему нам поручили написать отчёт, с отчётом трудностей возникнуть у нас не должно, самое главное оформить все по плану.