Практическая работа№1.

1. Напишите программу, которая позволяет ввести координаты точки и применить к ним матричное преобразование с матрицей:

$$T = \begin{pmatrix} 1 & 3 \\ 4 & 1 \end{pmatrix}$$

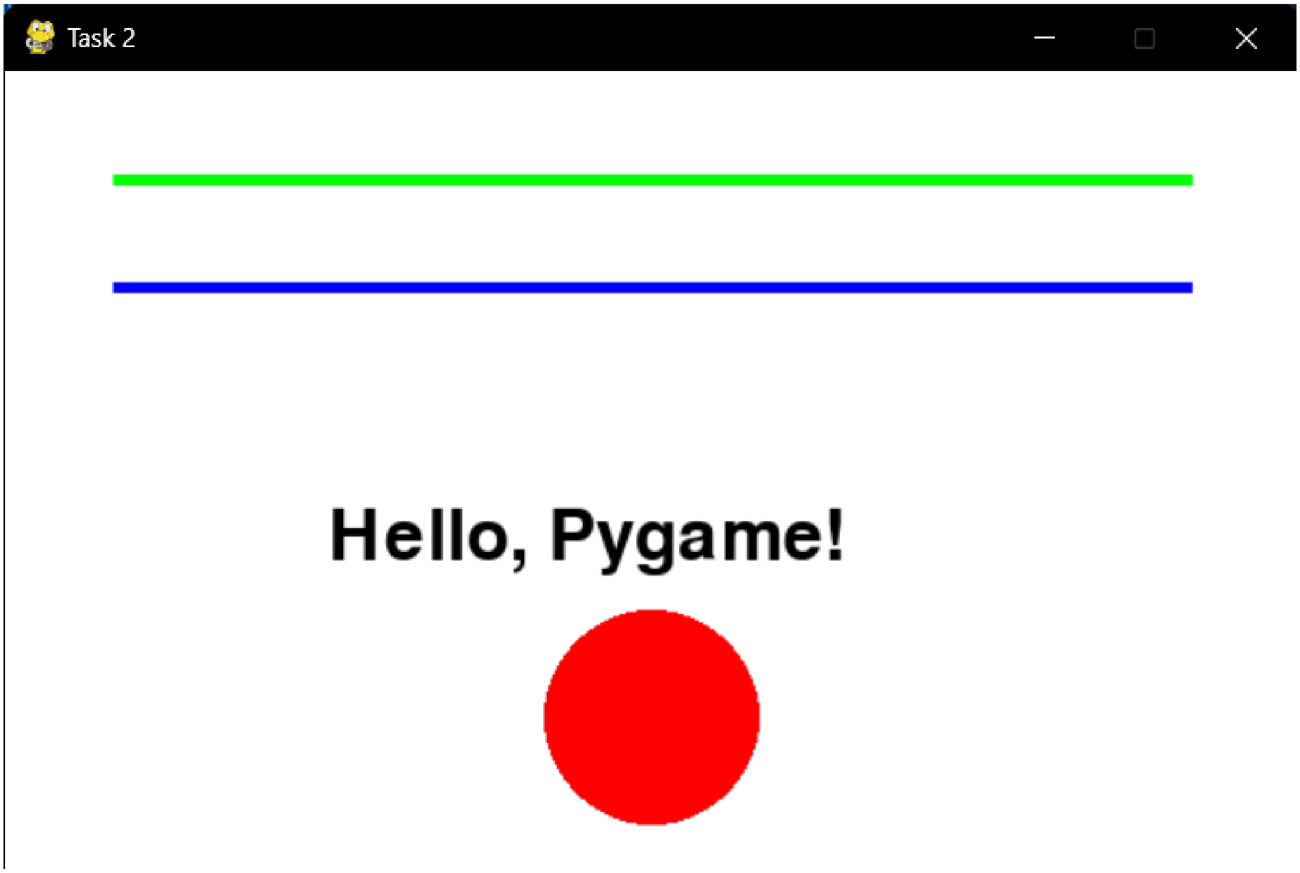
Выведите на экран начальные и полученные координаты. Прорисуйте исходную и новую точки в окне с использованием `pygame`.

Решение:


```
pygame 2.6.1 (SDL 2.28.4, Python 3.13.0)
Hello from the pygame community. https://www.pygame.org/contribute.html
Начальные координаты: [10 20]
Преобразованные координаты: [70 60]
```

2. Нарисуйте графические примитивы (`pygame`) — окружность, линии, текст — различными цветами в окне программы.

Решение:

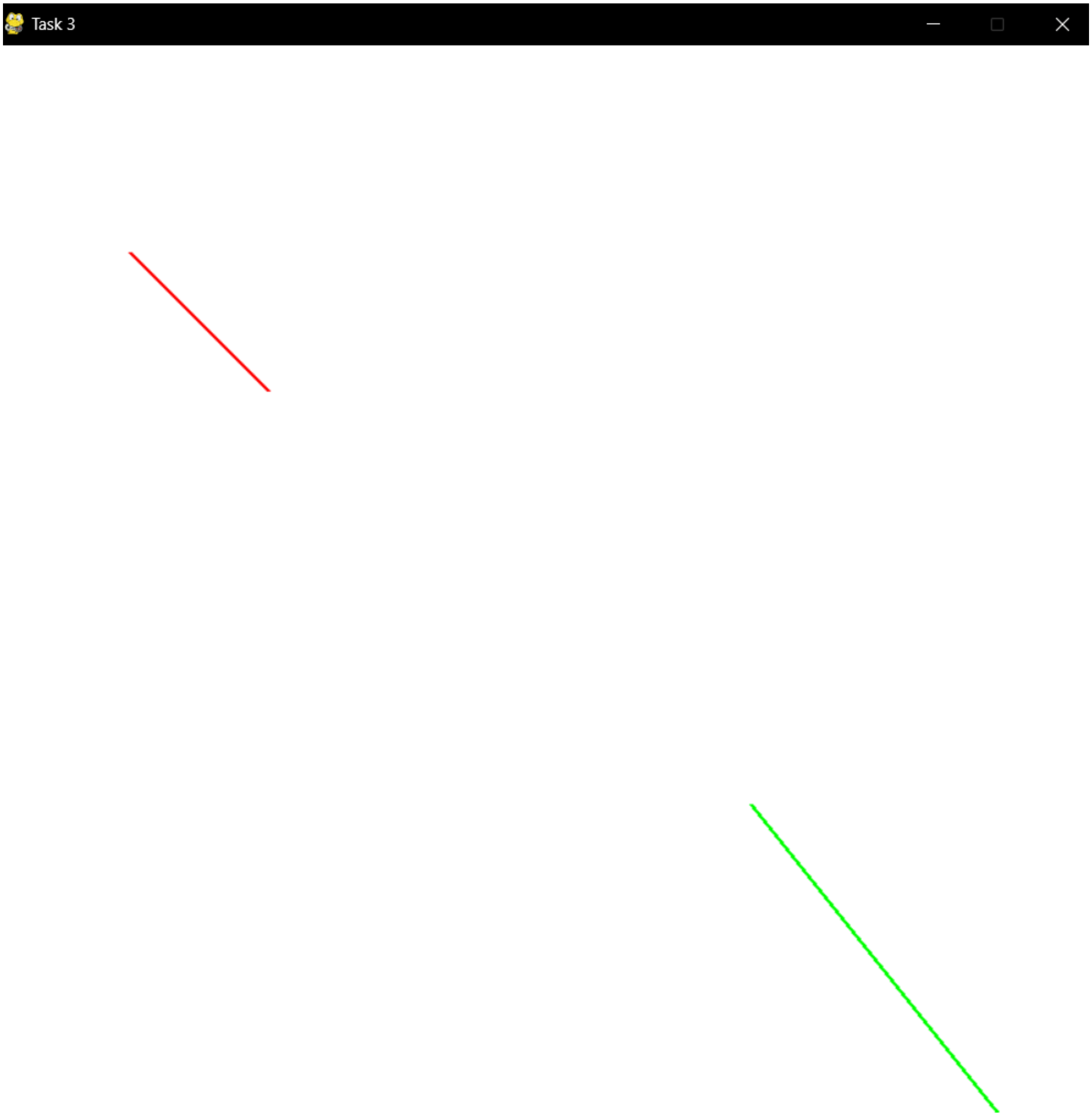


3. Напишите программу, которая преобразует отрезок, заданный двумя точками. Запишите координаты в матрицу отрезка и примените к ней матричное преобразование:

$$T = \begin{pmatrix} 1 & 3 \\ 4 & 1 \end{pmatrix}$$

Отобразите исходный и преобразованный отрезок разными цветами.

Решение:


Task 3

4. Напишите программу для преобразования координат конца отрезка

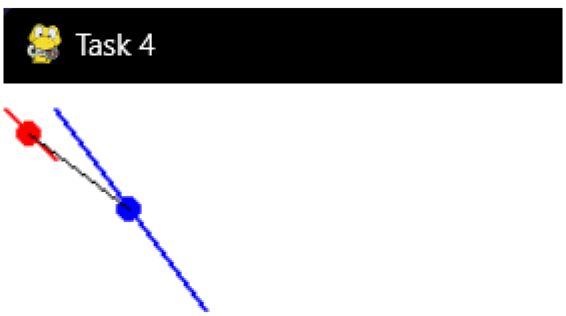
$$L = \begin{pmatrix} 0 & 100 \\ 200 & 300 \end{pmatrix}$$

по матрице:

$$T = \begin{pmatrix} 1 & 2 \\ 3 & 1 \end{pmatrix}$$

Найдите середину исходного и преобразованного отрезка и визуализируйте оба отрезка с помощью `pygame`. Обозначьте середины отрезков небольшими кругами и соедините их ещё одним отрезком.

Решение:

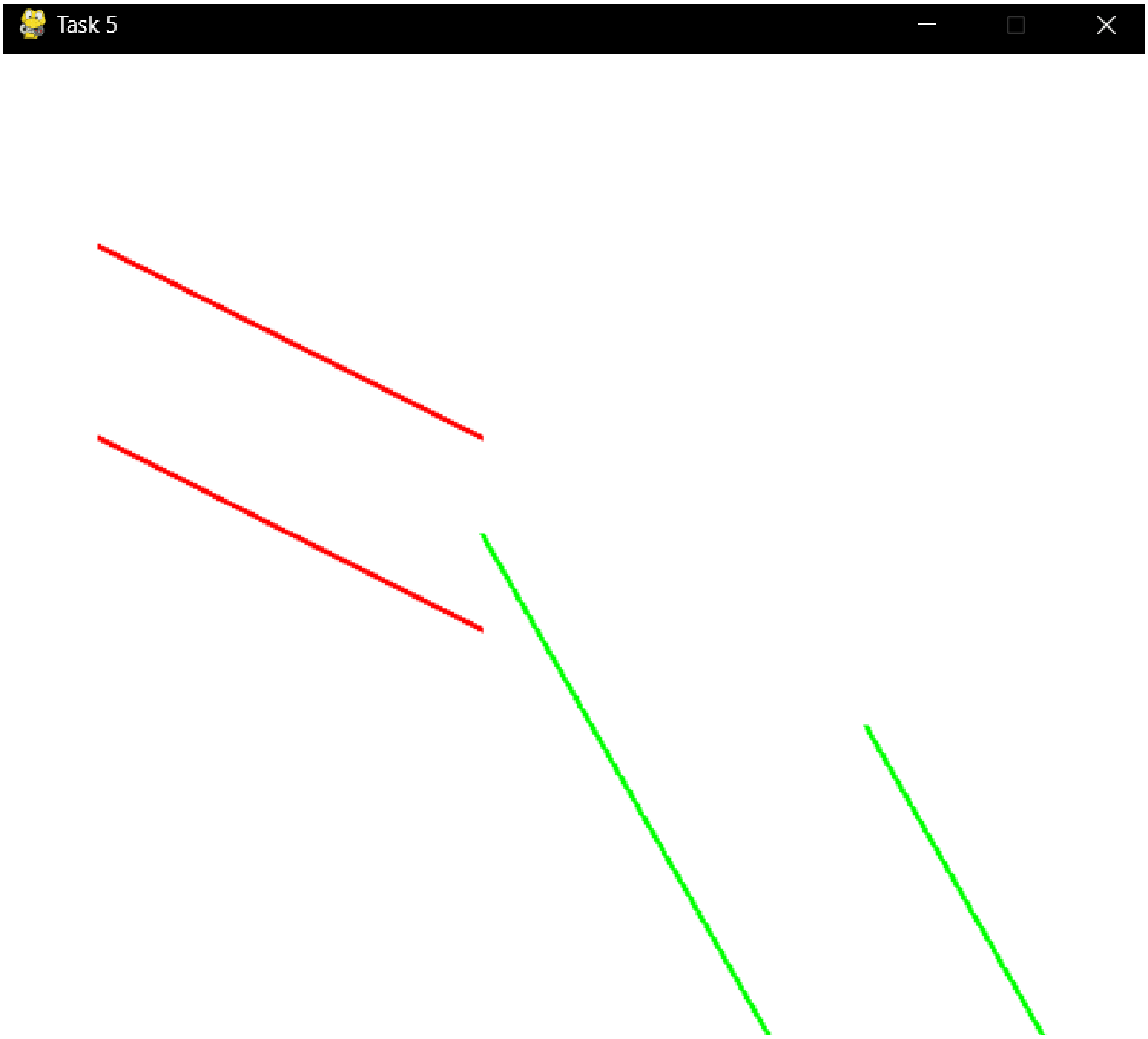


5. Преобразуйте два параллельных отрезка, заданных матрицей

$$L = \begin{pmatrix} 50 & 100 \\ 250 & 200 \\ 50 & 200 \\ 250 & 300 \end{pmatrix},$$

с помощью той же матрицы T, что и в предыдущей задаче. Рассчитайте и отобразите их начальный и конечный наклоны.

Решение:



6. Напишите программу для преобразования пересекающихся отрезков

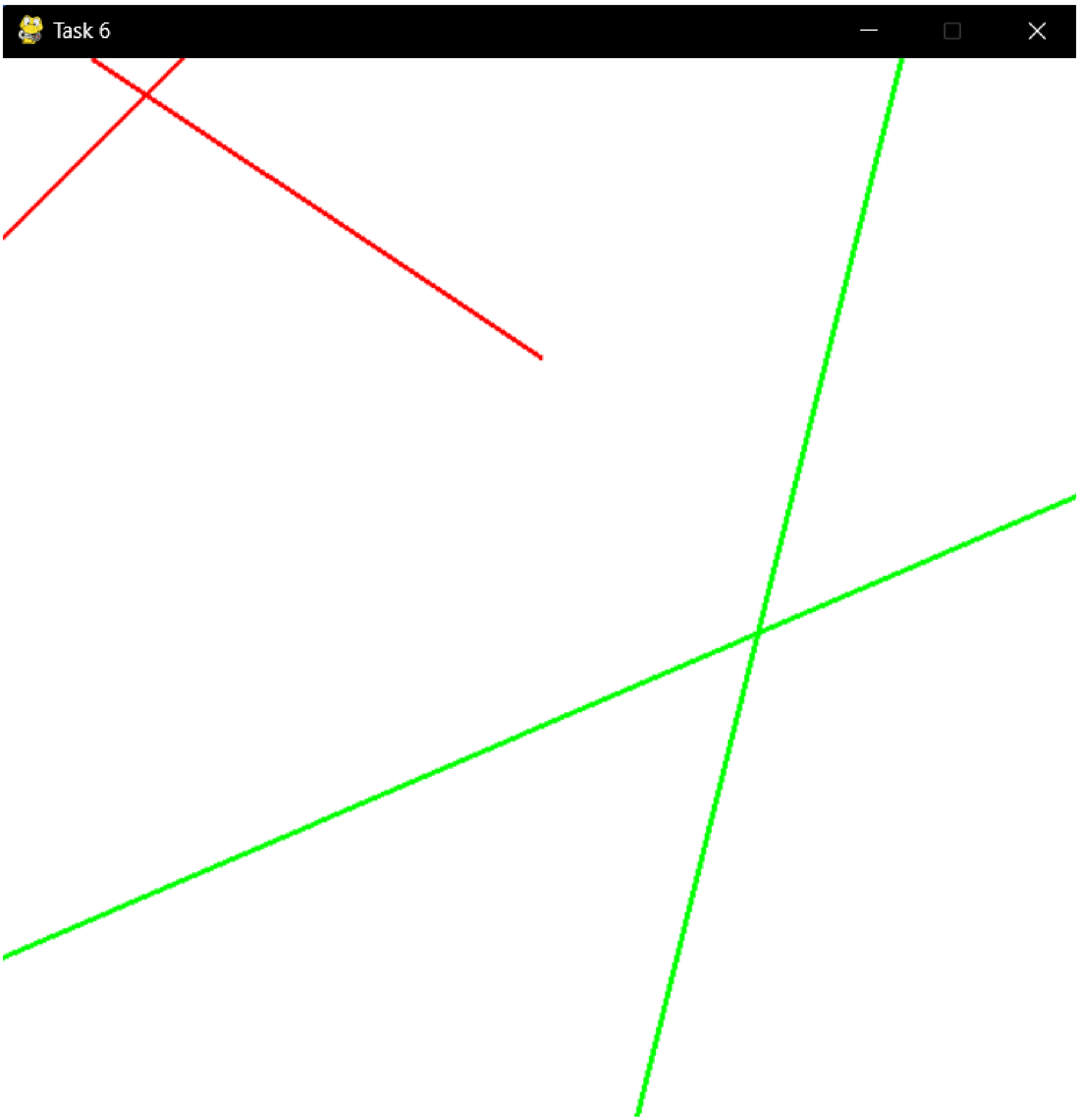
$$L = \begin{pmatrix} -1/2 & 3/2 \\ 3 & -2 \\ -1 & -1 \\ 3 & 5/3 \end{pmatrix}$$

по матрице:

$$T = \begin{pmatrix} 1 & 2 \\ 1 & -3 \end{pmatrix}$$

Искусственно сместите все получившиеся отрезки в видимую область игрового окна для наглядности (сохраняйте оригинальную матрицу координат в изначальной матрице L !). Перед преобразованиями умножьте искусственно матрицу отрезка L на 100, чтобы он и его преобразованная версия имели видимую длину в пикселях.

Решение:



7. Вращайте треугольник

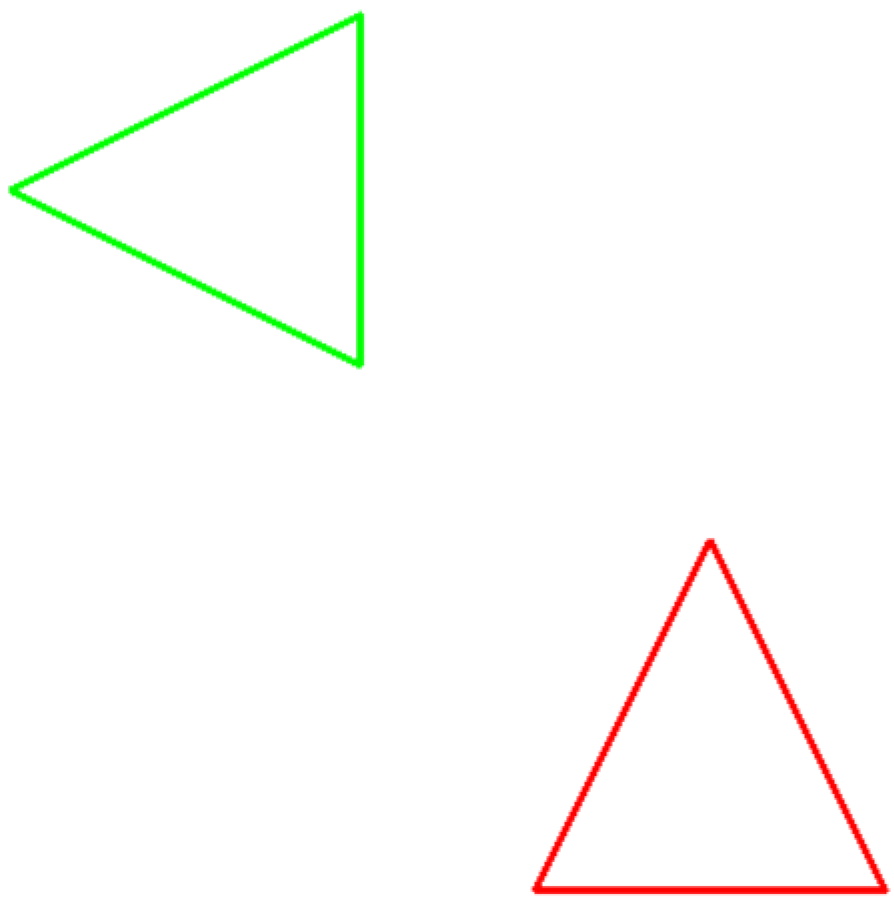
$$L = \begin{pmatrix} 3 & -1 \\ 4 & 1 \\ 2 & 1 \end{pmatrix}$$

на 90 градусов ($\pi/2$) против часовой стрелки с помощью матрицы:

$$T = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

Перед преобразованиями умножьте искусственно матрицу отрезка L на 100, чтобы он и его преобразованная версия имели видимую длину в пикселях. Как и в предыдущей задаче, смещайте оригинальный треугольник, чтобы он был виден на экране (сохраняйте оригинальную матрицу координат в изначальной матрице L !), после преобразования матрицы L сместите на такое же количество пикселей в видимую область окна новый треугольник L^*.

Решение:



8. Отразите треугольник

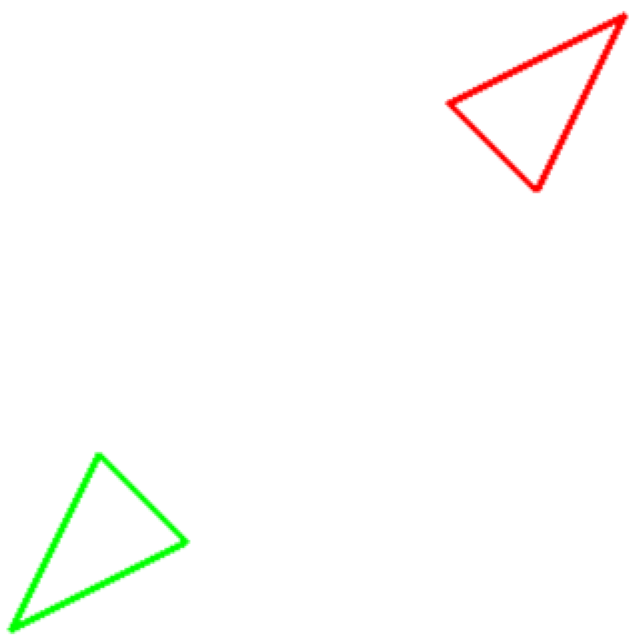
$$L = \begin{pmatrix} 8 & 1 \\ 7 & 3 \\ 6 & 2 \end{pmatrix}$$

относительно линии $y = x$ с помощью матрицы:

$$T = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Также прибегайте к смещениям в пикселях, к умножению на 100 и тому подобным приёмам, как и в предыдущей задаче, чтобы треугольники были видны.

Решение:



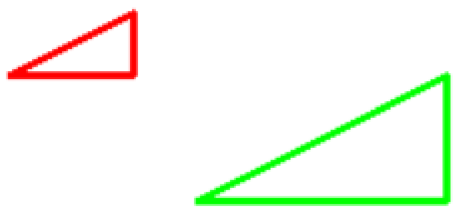
9. Масштабируйте (с надлежащим изменением координат в пикселях, чтобы всё было видно) треугольник

$$L = \begin{pmatrix} 5 & 1 \\ 5 & 2 \\ 3 & 2 \end{pmatrix}$$

с помощью матрицы:

$$T = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$

Решение:

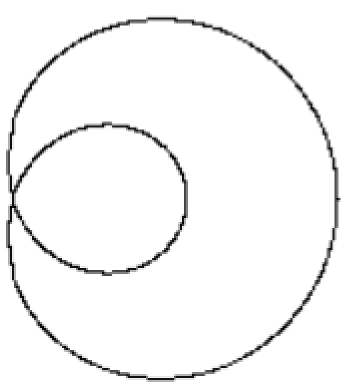


10. Нарисуйте на экране спираль в виде улитки Паскаля, используя полярные координаты:

$$r = b + 2 \cdot a \cdot \cos(\theta)$$

$$x = r \cdot \cos(\theta), \quad y = r \cdot \sin(\theta)$$

Решение:



11. Постройте квадрат, масштабируйте его с коэффициентом $m = 0.9$ и поворачивайте на угол $\alpha = \pi/32$. Начальные координаты квадрата:

$$X = \begin{pmatrix} 2 & -2 \\ -2 & -2 \\ -2 & 2 \\ 2 & 2 \end{pmatrix} \times 100$$

Используйте комбинированное преобразование и выполните 20 таких операций с отрисовкой в `pygame`.

Решение (квадрат динамически уменьшается с разворотом по часовой стрелке):