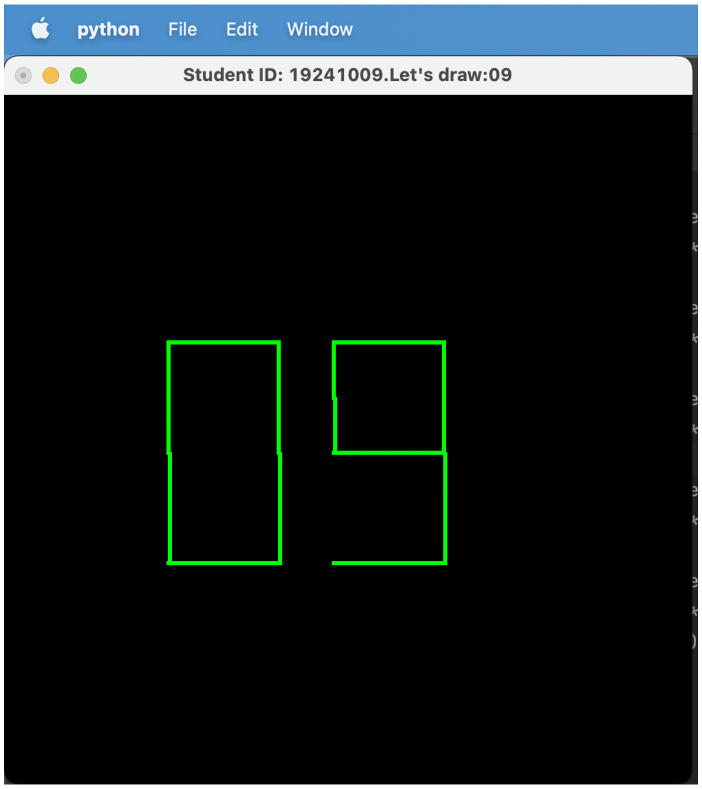
```
from OpenGL.GL import *
from OpenGL.GLUT import *
from OpenGL.GLU import *

def find_zone(x1,y1,x2,y2):
    dx = x2 - x1
    dy = y2 - y1
    zone = 0

    if abs(dx) > abs(dy):
        if dx > 0 and dy > 0:
            zone = 0
        elif dx < 0 and dy > 0:
            zone = 3
        elif dx > 0 and dy < 0:
            zone = 7
        elif dx < 0 and dy < 0:
            zone = 4

    else:
        if dx > 0 and dy > 0:
            zone = 1
        elif dx < 0 and dy > 0:
            zone = 2
        elif dx > 0 and dy < 0:
            zone = 6
        elif dx < 0 and dy < 0:
            zone = 5
    return zone

def convert_to_zone0(x,y,zone):
    if zone== 1:
        converted_x = y
        converted_y = x
    elif zone== 2:
        converted_x = y
        converted_y = -x
    elif zone== 3:
        converted_x = -x
        converted_y = y
    elif zone==4:
        converted_x = -x
        converted_y = -y
    elif zone==5:
        converted_x = -y
        converted_y = -x
    elif zone==6:
        converted_x = -y
        converted_y = x
    else:
        converted_x = x
        converted_y = -y

    return converted_x, converted_y

def revert(x,y,zone):
    if zone== 1:
        converted_x = y
        converted_y = x
    elif zone==2:
        converted_x = -y
        converted_y = x
    elif zone==3:
        converted_x = -x
        converted_y = y
    elif zone==4:
        converted_x = -x
        converted_y = -y
    elif zone==5:
        converted_x = -y
        converted_y = -x
    elif zone==6:
        converted_x = y
        converted_y = -x
    else:
        converted_x = x
        converted_y = -y
    return converted_x, converted_y

def midpoint_line(x1,y1,x2,y2):

    pixel_list = []
    dx = x2 - x1
    dy = y2 - y1
    d = (2 * dy) - dx
    pixel_list.append((x1,y1))
    x1+=1
    while x1 <= x2:
        if d > 0:
            d = d + 2 * (dy - dx)
            y1 = y1 + 1
        else:
            d = d + (2 * dy)
        pixel_list.append((x1, y1))
        x1+=1
    return pixel_list

def generate_pixels(x1,y1,x2,y2):
    zone = find_zone(x1,y1,x2,y2)
    if zone == 0:
        pixels = midpoint_line(x1,y1,x2,y2)
    else:
        converted1 = convert_to_zone0(x1,y1, zone)
        converted2 = convert_to_zone0(x2,y2, zone)
        pixels = midpoint_line(*converted1,*converted2)
        reverted = []
        for i in range(len(pixels)):
            reverted.append(revert(pixels[i][0], pixels[i][1], zone))
        pixels = reverted
    return pixels

def draw_points(x, y):
    glPointSize(3)
    glBegin(GL_POINTS)
    glVertex2f(x,y)
    glEnd()


def iterate():
    glViewport(0, 0, 400, 400)
    glMatrixMode(GL_PROJECTION)
    glLoadIdentity()
    glOrtho(0.0, 500, 0.0, 500, 0.0, 1.0)
    glMatrixMode (GL_MODELVIEW)
    glLoadIdentity()

def showScreen():
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
    glLoadIdentity()
    iterate()
    glColor3f(0.0, 1.0, 0.0)
    #0
    generated_pixels = generate_pixels(250, 400, 251, 200)
    for i in generated_pixels:
        draw_points(*i)
    generated_pixels = generate_pixels(150, 400, 250, 401)
    for i in generated_pixels:
        draw_points(*i)
    generated_pixels = generate_pixels(150, 400, 151, 200)
    for i in generated_pixels:
        draw_points(*i)
    generated_pixels = generate_pixels(150, 200, 250, 201)
    for i in generated_pixels:
        draw_points(*i)
    # 9
    generated_pixels = generate_pixels(400, 400, 401, 200)
    for i in generated_pixels:
        draw_points(*i)
    generated_pixels = generate_pixels(300, 400, 400, 401)
    for i in generated_pixels:
        draw_points(*i)
    generated_pixels = generate_pixels(300, 300, 400, 301)
    for i in generated_pixels:
        draw_points(*i)
    generated_pixels = generate_pixels(300, 200, 400, 201)
    for i in generated_pixels:
        draw_points(*i)
    generated_pixels = generate_pixels(300, 400, 301, 300)
    for i in generated_pixels:
        draw_points(*i)
    glutSwapBuffers()



glutInit()
glutInitDisplayMode(GLUT_RGBA)
glutInitWindowSize(500, 500)
glutInitWindowPosition(0, 0)
wind = glutCreateWindow("Student ID: 19241009.Let's draw:09")
glutDisplayFunc(showScreen)
```

```
glutMainLoop()
```


python
File
Edit
Window
Student ID: 19241009.Let's draw:09