**Code**

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

found_zone = None

def find_zone(x1, y1, x2, y2):
    dx = x2 - x1
    dy = y2 - y1
    zone = None
    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 = 4
        elif dx >= 0 and dy <= 0:
            zone = 7
    else:
        if dx >= 0 and dy >= 0:
            zone = 1
        elif dx <= 0 and dy >= 0:
            zone = 2
        elif dx <= 0 and dy <= 0:
            zone = 5
        else: # dx= > 0 and dy <= 0
            zone = 6

    return zone


def convert_zone(x1, y1, x2, y2):
    global found_zone
    found_zone = find_zone(x1, y1, x2, y2)
    # print(found_zone)

    if found_zone == 0:
```

```
    return x1, y1, x2, y2
if found_zone == 1:
    a1 = y1
    b1 = x1
    a2 = y2
    b2 = x2
elif found_zone == 2:
    a1 = y1
    b1 = -x1
    a2 = y2
    b2 = -x2
elif found_zone == 3:
    a1 = -x1
    b1 = y1
    a2 = -x2
    b2 = y2
elif found_zone == 4:
    a1 = -x1
    b1 = -y1
    a2 = -x2
    b2 = -y2
elif found_zone == 5:
    a1 = -y1
    b1 = -x1
    a2 = -y2
    b2 = -x2
elif found_zone == 6:
    a1 = -y1
    b1 = x1
    a2 = -y2
    b2 = x2
else:
    a1 = x1
    b1 = -y1
    a2 = x2
    b2 = -y2

return a1, b1, a2, b2
```

```
def convert_to_origin(x1, y1):
    if found_zone == 0:
        return x1, y1

    if found_zone == 1:
        a1 = y1
        b1 = x1
    elif found_zone == 2:
        a1 = -y1
        b1 = x1
    elif found_zone == 3:
        a1 = -x1
        b1 = y1
    elif found_zone == 4:
        a1 = -x1
        b1 = -y1
    elif found_zone == 5:
        a1 = -y1
        b1 = -x1
    elif found_zone == 6:
        a1 = y1
        b1 = -x1
    else:
        a1 = x1
        b1 = -y1

    return a1, b1


def mp_l(a1, b1, a2, b2):
    x1, y1, x2, y2 = convert_zone(a1, b1, a2, b2)
    dx = x2 - x1
    dy = y2 - y1
    d = 2 * dy - dx
    dNE = 2 * (dy - dx)
    dE = 2 * dy

    x = x1
    y = y1
    while x < x2:
```

```
        a, b = convert_to_origin(x, y)
        draw_points(a, b)
        if d <= 0:
            d = d + dE
            x += 1
            y += 0
            # print('d <= 0', d, x, y)
        else:
            d = d + dNE
            x += 1
            y += 1
            # print('d > 0', d, x, y)


def digit_to_pix(pos, num):
    if num == 0:
        if pos == 1:
            return (-50, 200, -300, 200), (-300, 200, -300, -200), (-300, -200, -50, -200), (-50,
-200, -50, 200)
        return (50, 200, 300, 200), (300, 200, 300, -200), (300, -200, 50, -200), (50, -200,
50, 200)
    elif num == 1:
        if pos == 1:
            return (-50, 200, -50, -200)
        return (300, 200, 300, -200)
    elif num == 2:
        if pos == 1:
            return (-300, 200, -50, 200), (-50, 200, -50, 0), (-50, 0, -300, 0), (-300, 0, -300,
-200), (-300, -200, -50, -200)
        return (50, 200, 300, 200), (300, 200, 300, 0), (300, 0, 50, 0), (50, 0, 50, -200), (50,
-200, 300, -200)
    elif num == 3:
        if pos == 1:
            return (-300, 200, -50, 200), (-50, 200, -50, 0), (-50, 0, -300, 0), (-50, 0, -50,
-200), (-50, -200, -300, -200)
        return (50, 200, 300, 200), (300, 200, 300, 0), (300, 0, 50, 0), (300, 0, 300, -200),
(300, -200, 50, -200)
    elif num == 4:
        if pos == 1:
            return (-50, 200, -50, -200), (-50, 0, -300, 0), (-300, 0, -300, 200)
```

```
        return (50, 0, 50, 200), (50, 0, 300, 0), (300, 200, 300, -200)
    elif num == 5:
        if pos == 1:
            return (-300, 200, -50, 200), (-300, 200, -300, 0), (-300, 0, -50, 0), (-50, 0, -50,
-200), (-50, -200, -300, -200)
        return (50, 200, 300, 200), (50, 200, 50, 0), (50, 0, 300, 0), (300, 0, 300, -200),
(300, -200, 50, -200)
    elif num == 6:
        if pos == 1:
            return (-50, 200, -300, 200), (-300, 200, -300, -200), (-300, -200, -50, -200), (-50,
-200, -50, 0), (-50, 0, -300, 0)
        return (300, 200, 50, 200), (50, 200, 50, -200), (50, -200, 300, -200), (300, -200,
300, 0), (300, 0, 50, 0)
    elif num == 7:
        if pos == 1:
            return (-50, 200, -50, -200), (-50, 200, -300, 200)
        return (50, 200, 300, 200), (300, 200, 300, -200)
    elif num == 8:
        if pos == 1:
            return (-50, 200, -300, 200), (-300, 200, -300, -200), (-300, -200, -50, -200), (-50,
-200, -50, 0), (-50, 0, -300, 0), (-50, 0, -50, 200)
        return (300, 200, 50, 200), (50, 200, 50, -200), (50, -200, 300, -200), (300, -200,
300, 0), (300, 0, 50, 0), (300, 0, 300, 200)
    elif num == 9:
        if pos == 1:
            return (-50, 200, -50, -200), (-50, 200, -300, 200),  (-300, 200, -300, 0), (-300, 0,
-50, 0)
        return (50, 200, 300, 200), (300, 200, 300, -200), (50, 200, 50, 0), (50, 0, 300, 0)


def draw_points(x, y):
    glPointSize(10) #pixel size. by default 1 thake
    glBegin(GL_POINTS)
    glVertex2f(x,y) #jekhane show korbe pixel
    glEnd()

def iterate():
    glViewport(0, 0, 500, 500)
    glMatrixMode(GL_PROJECTION)
    glLoadIdentity()
```

```
    glOrtho(-500.0, 500, -500.0, 500, 0.0, 1.0)
    glMatrixMode(GL_MODELVIEW)
    glLoadIdentity()

def showScreen():
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
    glLoadIdentity()
    iterate()
    glColor3f(0.4, 0.9, 0.6) #konokichur color set (RGB)

    # My code starts here
=================================================================
=============

    ui = input("Enter your student ID: ")     # My student ID is 19101147
    tup_list = []
    for i in range(1, 3):
        digit = int(ui[-3+i])
        tup_list.append(digit_to_pix(i, digit))

    for tup in tup_list:
        if isinstance(tup[0], tuple):
            for t in tup:
                mp_l(*t)
        else:
            mp_l(*tup)

    glutSwapBuffers()


glutInit()
glutInitDisplayMode(GLUT_RGBA)
glutInitWindowSize(500, 500) #window size
glutInitWindowPosition(0, 0)
wind = glutCreateWindow(b"Lab 2")  #window name
glutDisplayFunc(showScreen)

glutMainLoop()
```

**Output**