```
import matplotlib.pyplot as plt

import random


# Chaos Game Algorithm!


# Vertices of an equilateral triangle

vertices = [(0, 0), (1, 0), (0.5, 0.866)]

# A function to check whether point (x, y)

# lies inside the triangle formed by

# A0, 0), (1, 0) and (0.5, 0.866)


def isInside(x1, y1, x2, y2, x3, y3, x, y):

    def area(x1, y1, x2, y2, x3, y3):

        return abs((x1*(y2 - y3) + x2*(y3 - y1) + x3*(y1 - y2)) / 2.0)


    A = area(x1, y1, x2, y2, x3, y3)

    A1 = area(x, y, x2, y2, x3, y3)

    A2 = area(x1, y1, x, y, x3, y3)
```

```
    A3 = area(x1, y1, x2, y2, x, y)


    return A == A1 + A2 + A3


# Prompting the user for a seed point
while True:
    try:
        seed_x = float(input("Enter the x-coordinate of the seed point: "))
        seed_y = float(input("Enter the y-coordinate of the seed point: "))
        if isInside(0, 0, 1, 0, 0.5, 0.866, seed_x, seed_y):
            print("Valid seed point entered.")
            break
        else:
            print("The point is not inside the triangle. Please try again.")
    except ValueError:
        print("Invalid input. Please enter numerical values.")


#This is your starting point
```

```
seed=(seed_x, seed_y)


# list where points are stored (x_t,y_t),
# starting with your seed
points = [seed]


# Prompting user for the number of steps
while True:
    try:
        num_steps = int(input("Enter the number of steps: "))
        if num_steps > 0:
            print(f"Number of steps set to {num_steps}.")
            break
        else:
            print("Please enter a positive integer.")
    except ValueError:
        print("Invalid input. Please enter a positive integer.")
```

```
for i in range(num_steps):

    # Choosing a random vertex to move toward from the list 'vertices'

    pt_vertex = vertices[random.randint(0,2)]


    #Next point made by moving from the last point to the midpoint

    next_point= ((points[-1][0] + pt_vertex[0]) / 2, (points[-1][1] + pt_vertex[1]) / 2)


    #add the new point to list of points

    points.append(next_point)


# Plotting points

plt.scatter([p[0] for p in points], [p[1] for p in points], s=0.1)

plt.show()
```

```
Enter the x-coordinate of the seed point:  0
Enter the y-coordinate of the seed point:  0
Valid seed point entered.
Enter the number of steps:  10000
Number of steps set to 10000.
```