# Question 3

# Recursive function to generate a geometric Fractal pattern using Python's turtle graphics (Code Explanation)

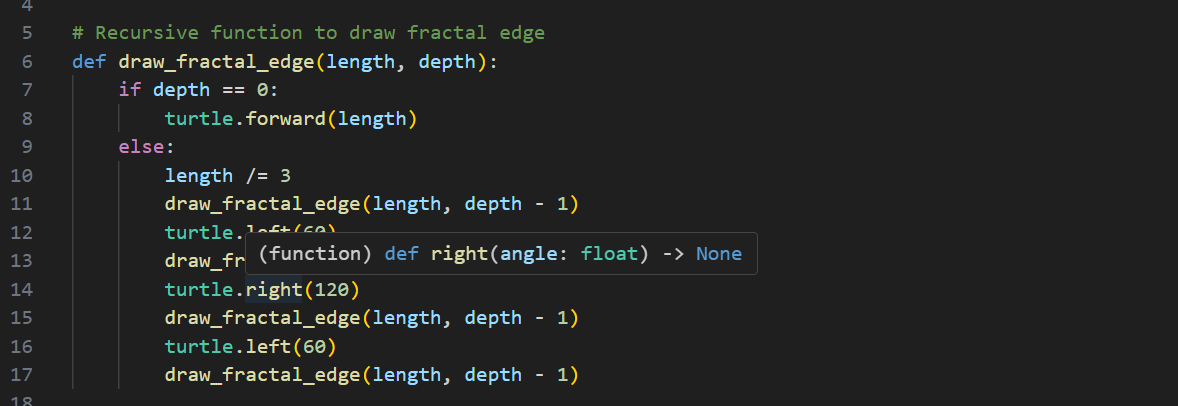
This program starts with a regular polygon and creates a fractal pattern using recursion and Python’s turtle graphics. Like the Koch snowflake, the edges of the polygon are modified recursively to produce complex geometric patterns. The final output is a symmetrical, self-similar pattern that increases in complexity with each increase in recursion depth.

1. Importing the turtle Module



Loading Python’s built-in turtle graphics library. This will enable drawing shapes and patterns using a virtual “pen”.

1. Defining the recursive function



The function above is the core of the fractal generation process. Base case (depth ==0)

Draws a straight line of the given length. This represents the simplest form of edge without any fractal detail.

Recursive case:

* First else condition divides the line into three equal parts.
* Second condition replaces the middle part of the line with two sides of an equilateral triangle pointing inward.
* Third condition transform straight edges into four smaller segments.
* With depth – 1, each segment is recursively processed, generating more detail at each level.

This recursive transform creates the zigzag seen in the output image

1. Getting Users' Input

A screen shot of a computer program

AI-generated content may be incorrect.

This command prompts the users to enter the sides, side length, and depth of the fractal pattern.

The number of sides determines the shape of the fractal pattern, and depth controls how many times the indentation is applied.

1. Turtle Setup

A screen shot of a computer

AI-generated content may be incorrect.

The prompts set up the turtle to speed up the rendering, background colour, and set the pen colour to blue.

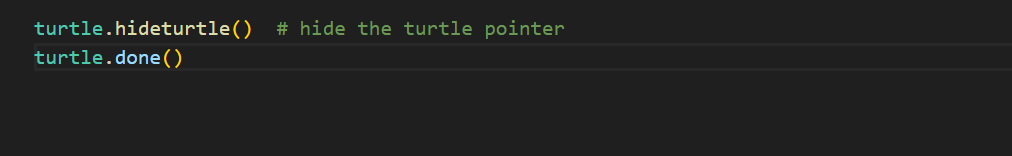
1. Drawing the fractal polygon

A black rectangle with white text

AI-generated content may be incorrect.

This loop makes sure the loop applies through each side of the polygon and rotates the turtle by 360/ sides to position it for the next side.

1. Finishing the Drawing



These prompts will hide the turtle cursor and end the turtle session, leaving a clean and finished fractal pattern on the screen.

# Summary

This Python program creates a fractal design based on the Koch snowflake using recursion and turtle graphics. Each edge of a regular polygon is first split into three segments, and the middle segment is swapped into two inward-pointing sides of an equilateral triangle. Each depth level designs a more complex indentation. The end product is a symmetrical fractal that graphically illustrates the geometric recursion.