

Tecnológico de Costa Rica

Proyecto Programado 2
Fractales

IC-1801
Taller de Programación
Escuela de Ingeniería en Computación

Prof. Mauricio Avilés

Estudiantes
Axel Fernández Jiménez
2016098894

María José Paz Gómez
2016106862

07/04/2016

Tabla de contenido

MANUAL DE USUARIO	3
PASOS INICIALES	3
TRIÁNGULO DE SIERPINSKI	4
Corta Cabezas	6
ALFOMBRA DE SIERPINSKI	8
Sol Eterno	10
HEXÁGONO DE SIERPINSKI	12
Círculo Saltarín	14
CURVA DE KOCH	16
Rompecabezas de Koch	18
CÍRCULO DE RAVEL	20
Circo de Colores	22
CURVA DE SATIE	24
Espiral de Satie	26
ESTRELLA DE DEBUSSY	28
ColoredStar	30

Manual de Usuario

Se han creado un conjunto de programas para formar diferentes figuras. Este documento es para explicar el funcionamiento de cada uno de ellas, de forma detallada para que se eviten equivocaciones durante el proceso.

Para poder ejecutar los programas de la manera correcta, se deben realizar un conjunto de pasos:

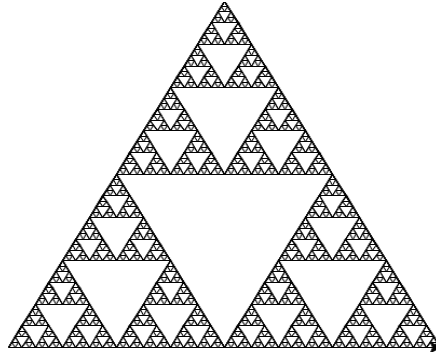
Pasos Iniciales

1. Lo primero que se debe hacer, es descargar Python, ya que es en este lenguaje en el que se programo la información.
2. Para que Python se pueda utilizar, se debe descargar IDLE, con esta aplicación se van a poder correr los programas, de lo contrario no se podrá utilizar el documento de Python brindado.
3. Para poder usar el documento de Python que contiene los programas, se tiene que guardar en la computadora

El documento de Fractales contiene varios programas, cada uno tiene su nombre en rojo y para separarlos se utilizo un conjunto de líneas (-) entre el signo de número (#). Además, cada uno de los programas tiene definido su funcionalidad y también sus entradas, salidas y restricciones para tener una mejor comprensión de cada uno de ellos.

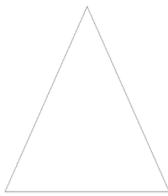
Triángulo de Sierpinski

El objetivo de este programa es lograr algo parecido al siguiente fractal:

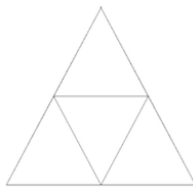


Consiste de un triángulo, en el cual su caso base es de profundidad uno, con esto se va a crear un triángulo normal, conforme se vaya aumentando la profundidad, se van a crear tres triángulos más dentro del triángulo anterior.

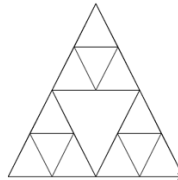
PROFUNDIDAD 1



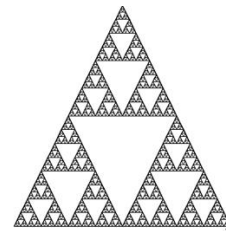
PROFUNDIDAD 2



PROFUNDIDAD 3



PROFUNDIDAD 7



Pasos

1. Se debe poner a correr el programa, esto se hace cuando ya se está en el documento de Fractales y se pulsa la tecla F5, esto hace que aparezca el shell de Python.
2. Para este programa, se debe poner en el primer renglón del shell de Python

```
>>> trianguloSierpinski(profundidad,lado)
```

3. Donde se menciona profundidad, se debe de colocar un número que defina la profundidad de la que desea que sea el triángulo, que va a decir la cantidad de triángulos que va a haber dentro del original, y cuantas veces va a realizar el proceso. El lado define de que tamaño va a ser el triángulo inicial. Éste parámetro también debe ser un número entero y positivo.

4. A continuación se presiona RETURN o ENTER y esto va a hacer que se abra una nueva ventana, la de turtle, donde va a aparecer el dibujo del fractal con los perímetros que se establecieron anteriormente.

Si hubo algún error durante el procedimiento anterior, el programa va a decir que deben ser números enteros mayores a 0.

```
>>> trianguloSierpinski(t, 2.5)  
Deben ser números enteros mayores que 0
```

Corta Cabezas

La manera creativa de demostrar el triángulo de Sierpinski es de la siguiente manera:



Una vez que se da la orden, aparece la imagen anterior, con el triángulo, y este se desplaza aleatoriamente por el fondo.

Pasos

1. Se debe poner a correr el programa, esto se hace cuando ya se está en el documento de Fractales y se pulsa la tecla F5, esto hace que aparezca el shell de Python.
2. Para este programa, se debe poner en el primer renglón del shell de Python

```
>>> trianguloSierpinskiMod(profundidad,lado)
```

3. Donde se menciona profundidad, se debe de colocar un número que defina la profundidad de la que desea que sea el triángulo, que va a decir la cantidad de triángulos que va a haber dentro del original, y cuantas veces va a realizar el proceso. El lado define de que tamaño va a ser el triángulo inicial. Éste parámetro también debe ser un número entero y positivo. **Se recomienda**
trianguloSierpinskiMod(2,100)

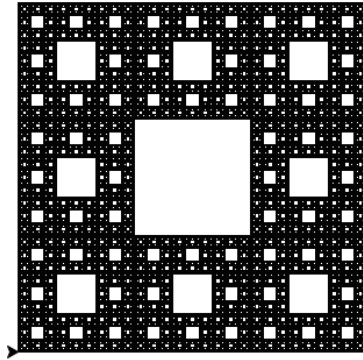
4. A continuación se presiona RETURN o ENTER y esto va a hacer que se abra una nueva ventana, la de turtle, donde va a aparecer el dibujo del fractal con los perímetros que se establecieron anteriormente.

Si hubo algún error durante el procedimiento anterior, el programa va a decir que deben ser números enteros mayores a 0.

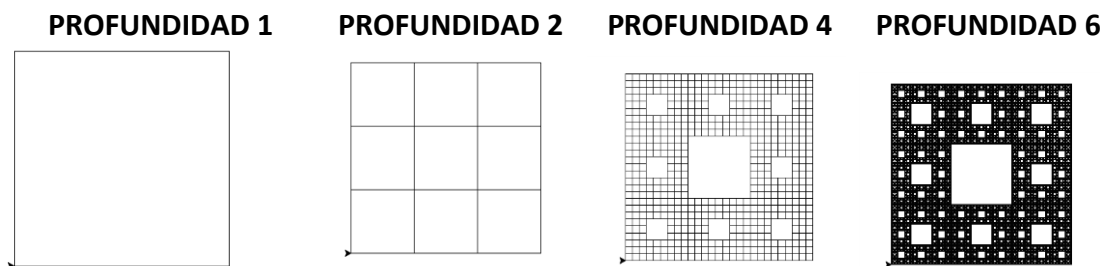
```
>>> trianguloSierpinskiMod(t, 2.5)  
Deben ser números enteros mayores que 0
```

Alfombra de Sierpinski

El objetivo de este programa es lograr algo parecido al siguiente fractal:



Consiste de un cuadrado, en el cual su caso base es de profundidad uno, con esto se va a crear un cuadrado normal, conforme se vaya aumentando la profundidad, el cuadrado se convierte en seis cuadrados más. Cada vez se deja el cuadrado del medio en blanco.



Pasos

1. Se debe poner a correr el programa, esto se hace cuando ya se está en el documento de Fractales y se pulsa la tecla F5, esto hace que aparezca el shell de Python.
2. Para este programa, se debe poner en el primer renglón del shell de Python

```
>>> alfombraSierpinski(profundidad,lado)
```

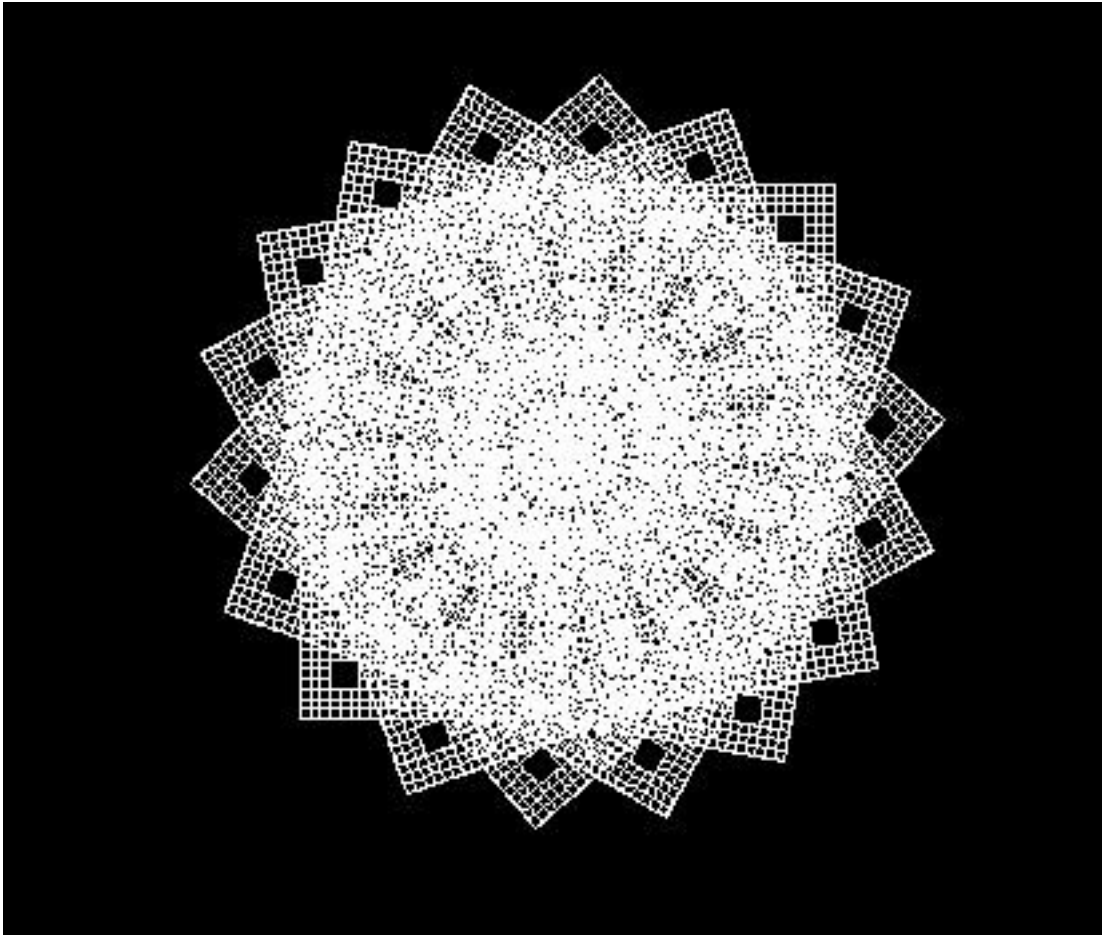

3. Donde se menciona profundidad, se debe de colocar un número que defina la profundidad de la que desea que sea el cuadrado, que va a decir la cantidad de cuadrados que va a haber dentro del original, y cuantas veces va a realizar el proceso. El lado define de que tamaño va a ser el cuadrado inicial. Éste parámetro también debe ser un número entero y positivo.
4. A continuación se presiona RETURN o ENTER y esto va a hacer que se abra una nueva ventana, la de turtle, donde va a aparecer el dibujo del fractal con los perímetros que se establecieron anteriormente.

Si hubo algún error durante el procedimiento anterior, el programa va a decir que deben ser números enteros mayores a 0.

```
>>> alfombraSierpinski(t, 2.5)  
Deben ser números enteros mayores que 0
```

Sol Eterno

La manera creativa de demostrar la alfombra de Sierpinski es de la siguiente manera:



Una vez que se da la orden, aparece la imagen anterior.

Pasos

1. Se debe poner a correr el programa, esto se hace cuando ya se está en el documento de Fractales y se pulsa la tecla F5, esto hace que aparezca el shell de Python.
2. Para este programa, se debe poner en el primer renglón del shell de Python

```
>>> alfombraSierpinskiMod(profundidad,lado)
```

3. Donde se menciona profundidad, se debe de colocar un número que defina la profundidad de la que desea que sea el fractal, que va a decir cuantas veces va a realizar el proceso. El lado define de que tamaño va a ser el fractal inicial. Éste parámetro también debe ser un número entero y positivo. **Se recomienda**
alfombraSierpinskiMod(4,100)
4. A continuación se presiona RETURN o ENTER y esto va a hacer que se abra una nueva ventana, la de turtle, donde va a aparecer el dibujo del fractal con los perímetros que se establecieron anteriormente.

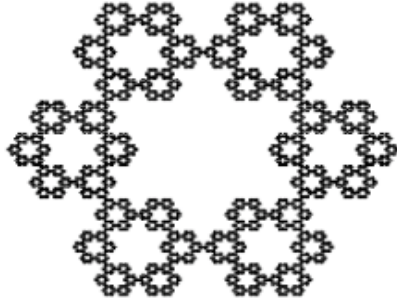
Si hubo algún error durante el procedimiento anterior, el programa va a decir que deben ser números enteros mayores a 0.

```
>>> alfombraSierpinskiMod(t, 2.5)
```

Deben ser números enteros mayores que 0

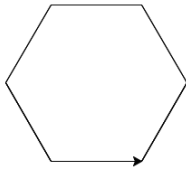
Hexágono de Sierpinski

El objetivo de este programa es lograr algo parecido al siguiente fractal:

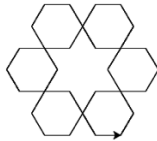


Consiste de un hexágono, en el cual su caso base es de profundidad uno, con esto se va a crear un hexágono normal, conforme se vaya aumentando la profundidad, se van a crear hexágonos de un tercio del tamaño sin que haya una línea que una los vértices de los hexágonos grandes.

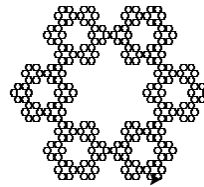
PROFUNDIDAD 1



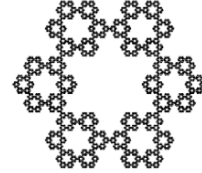
PROFUNDIDAD 2



PROFUNDIDAD 4



PROFUNDIDAD 6



Pasos

1. Se debe poner a correr el programa, esto se hace cuando ya se está en el documento de Fractales y se pulsa la tecla F5, esto hace que aparezca el shell de Python.
2. Para este programa, se debe poner en el primer renglón del shell de Python

```
>>> hexagonoSierpinski(profundidad,lado)
```

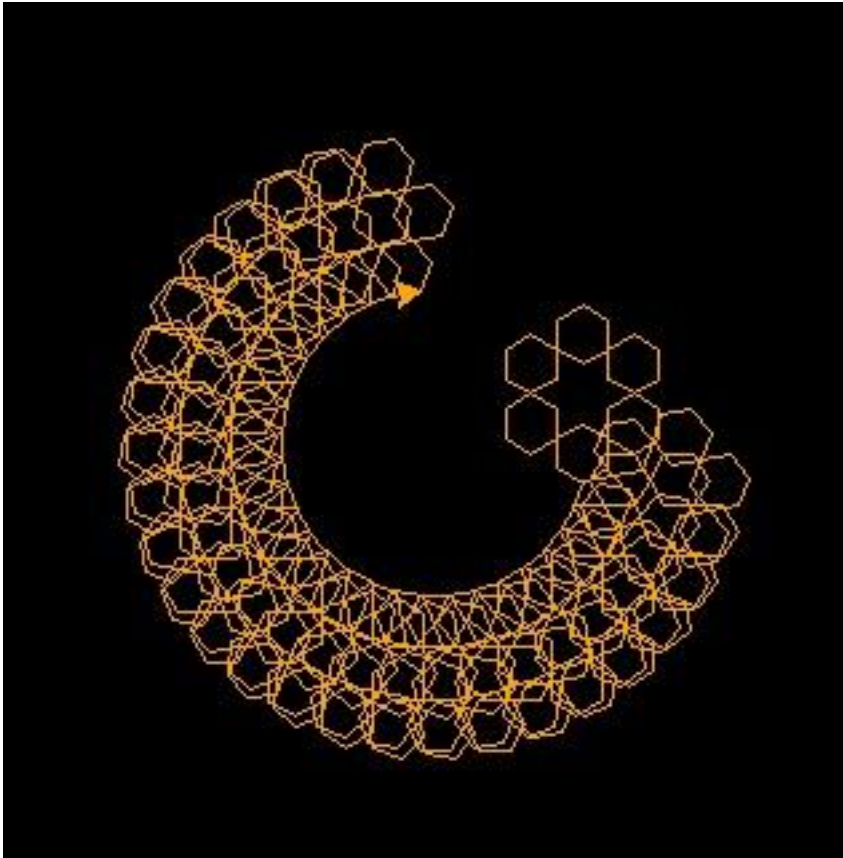
3. Donde se menciona profundidad, se debe de colocar un número que defina la profundidad de la que desea que sea el hexágono, que va a decir la cantidad de hexágonos que va a haber dentro del original, y cuantas veces va a realizar el proceso. El lado define de que tamaño va a ser el hexágono inicial. Éste parámetro también debe ser un número entero y positivo.
4. A continuación se presiona RETURN o ENTER y esto va a hacer que se abra una nueva ventana, la de turtle, donde va a aparecer el dibujo del fractal con los perímetros que se establecieron anteriormente.

Si hubo algún error durante el procedimiento anterior, el programa va a decir que deben ser números enteros mayores a 0.

```
>>> hexagonoSierpinski(t, 2.5)  
Deben ser números enteros mayores que 0
```

Círculo Saltarín

La manera creativa de demostrar el hexágono de Sierpinski es de la siguiente manera:



Una vez que se da la orden, aparece un solo hexágono, y se van creando más, uno a la par del otro creando un círculo continuo. Se completa y desaparece continuamente.

Pasos

1. Se debe poner a correr el programa, esto se hace cuando ya se está en el documento de Fractales y se pulsa la tecla F5, esto hace que aparezca el shell de Python.
2. Para este programa, se debe poner en el primer renglón del shell de Python

```
>>> hexagonoSierpinskiMod(profundidad,lado)
```

3. Donde se menciona profundidad, se debe de colocar un número que defina la profundidad de la que desea que sea el hexágono, que va a decir la cantidad de hexágonos que va a haber dentro del original, y cuantas veces va a realizar el proceso. El lado define de que tamaño va a ser el hexágono inicial. Éste parámetro también debe ser un número entero y positivo. **Se recomienda**
hexagonoSierpinskiMod(2,100)
4. A continuación se presiona RETURN o ENTER y esto va a hacer que se abra una nueva ventana, la de turtle, donde va a aparecer el dibujo del fractal con los perímetros que se establecieron anteriormente.

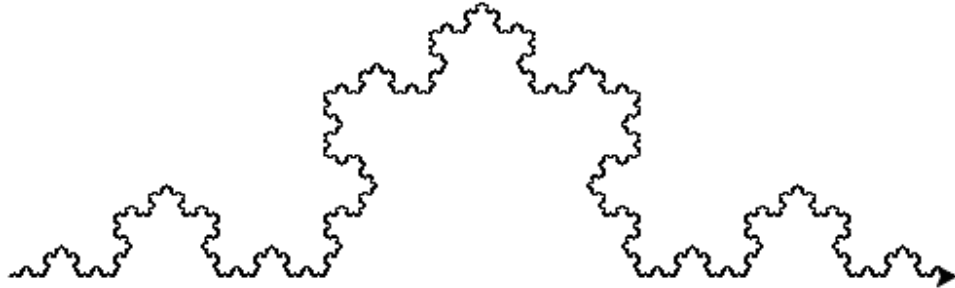
Si hubo algún error durante el procedimiento anterior, el programa va a decir que deben ser números enteros mayores a 0.

```
>>> hexagonoSierpinskiMod(t, 2.5)
```

Deben ser números enteros mayores que 0

Curva de Koch

El objetivo de este programa es lograr algo parecido al siguiente fractal:

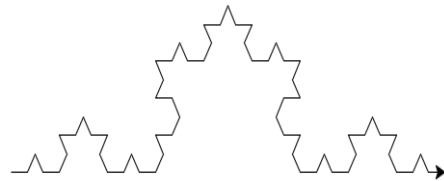


Consiste en dibujar una línea, y esta se divide en cuatro líneas de un tercio del tamaño original. Se dibuja una de las líneas del tercio de la original, después gira a la izquierda 60 grados, se dibuja otra línea del tercio de la original, se gira a la derecha 120 grados, otra línea mas del tercio de la original, gira a la izquierda 30 grados y se hace otra línea mas de tercio de la original. Las líneas que son el tercio de la original se hacen recursivamente.

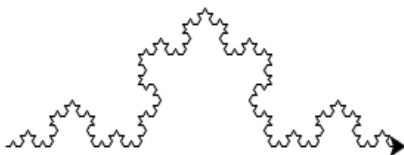
PROFUNDIDAD 1



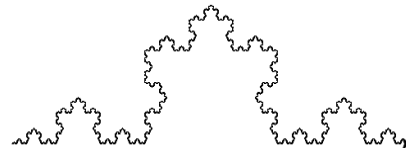
PROFUNDIDAD 3



PROFUNDIDAD 4



PROFUNDIDAD 6



Pasos

1. Se debe poner a correr el programa, esto se hace cuando ya se está en el documento de Fractales y se pulsa la tecla F5, esto hace que aparezca el shell de Python.
2. Para este programa, se debe poner en el primer renglón del shell de Python

```
>>> koch(profundidad,lado)
```

3. Donde se menciona profundidad, se debe de colocar un número que defina la profundidad de la que desea que sea el fractal, que va a decir la cantidad de veces que se va a repetir la recursión. El lado define de que tamaño va a ser la figura inicial. Éste parámetro también debe ser un número entero y positivo.
4. A continuación se presiona RETURN o ENTER y esto va a hacer que se abra una nueva ventana, la de turtle, donde va a aparecer el dibujo del fractal con los perímetros que se establecieron anteriormente.

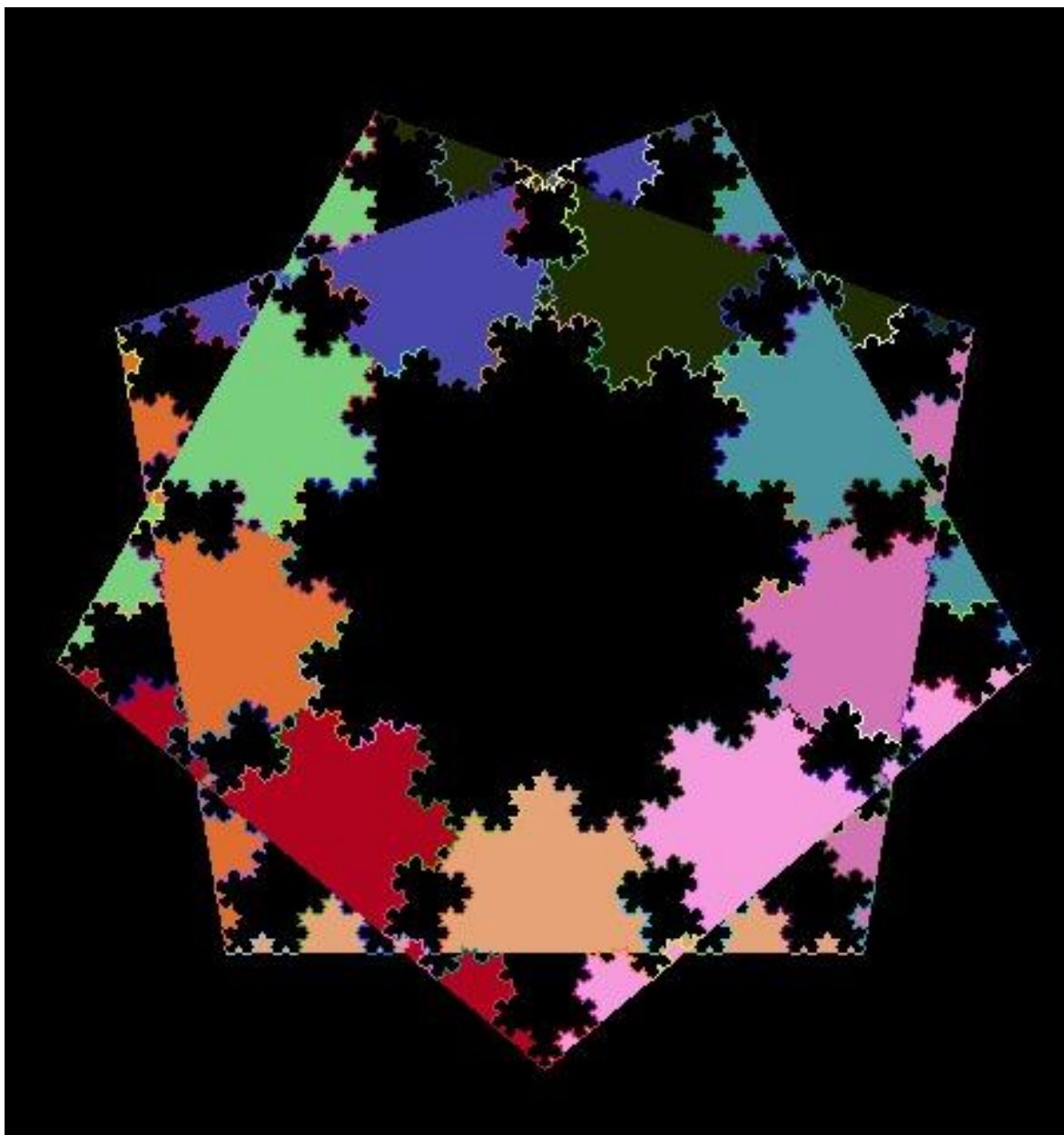
Si hubo algún error durante el procedimiento anterior, el programa va a decir que deben ser números enteros mayores a 0.

```
>>> koch(t, 2.5)
```

Deben ser números enteros mayores que 0

Rompecabezas de Koch

La manera creativa de demostrar la Curva de Koch es de la siguiente manera:



Una vez que se da la orden, aparecen un conjunto de Curvas de Koch con colores diferentes y realiza un conjunto de secuencias que forman una figura parecida a un rompecabezas, y se puede notar un copo grande en el centro de la figura.

Pasos

1. Se debe poner a correr el programa, esto se hace cuando ya se está en el documento de Fractales y se pulsa la tecla F5, esto hace que aparezca el shell de Python.
2. Para este programa, se debe poner en el primer renglón del shell de Python

```
>>> kochMod(profundidad,lado)
```

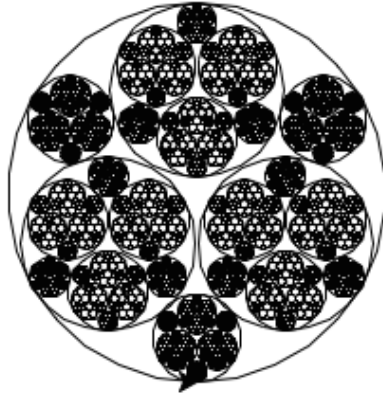
3. Donde se menciona profundidad, se debe de colocar un número que defina la profundidad de la que desea que sea el fractal, que va a decir la cantidad de veces que se va a repetir la recursión. El lado define de que tamaño va a ser la figura inicial. Éste parámetro también debe ser un número entero y positivo. **Se recomienda `kochMod(4,10)`**
4. A continuación se presiona RETURN o ENTER y esto va a hacer que se abra una nueva ventana, la de turtle, donde va a aparecer el dibujo del fractal con los perímetros que se establecieron anteriormente.

Si hubo algún error durante el procedimiento anterior, el programa va a decir que deben ser números enteros mayores a 0.

```
>>> kochMod (t, 2.5)  
Deben ser números enteros mayores que 0
```

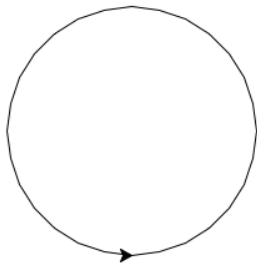
Círculo de Ravel

El objetivo de este programa es lograr algo parecido al siguiente fractal:

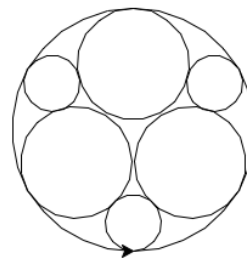


Consiste de un círculo, en el cual su caso base es de profundidad uno, con esto se va a crear un círculo normal, conforme se vaya aumentando la profundidad, se van a crear otros seis círculos, tres de ellos son $\frac{6}{26}$ partes del círculo original, y en los espacios en blanco que quedan van a haber tres círculos más de $\frac{46}{100}$ partes del original. Conforme vaya aumentando la profundidad, se va a repetir el proceso dentro de los otros círculos. El radio define de que tamaño es el círculo original.

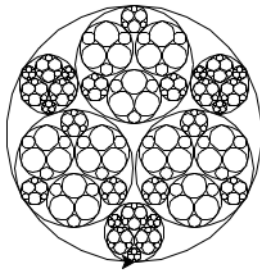
PROFUNDIDAD 1



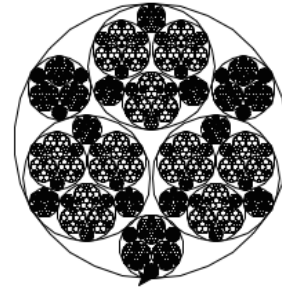
PROFUNDIDAD 2



PROFUNDIDAD 4



PROFUNDIDAD 6



Pasos

1. Se debe poner a correr el programa, esto se hace cuando ya se está en el documento de Fractales y se pulsa la tecla F5, esto hace que aparezca el shell de Python.
2. Para este programa, se debe poner en el primer renglón del shell de Python

```
>>> ravel(profundidad,radio)
```

3. Donde se menciona profundidad, se debe de colocar un número que defina la profundidad de la que desea que sea el fractal, que va a decir la cantidad de veces que se va a repetir la recursión. El radio define de que tamaño va a ser la figura inicial. Éste parámetro también debe ser un número entero y positivo.
4. A continuación se presiona RETURN o ENTER y esto va a hacer que se abra una nueva ventana, la de turtle, donde va a aparecer el dibujo del fractal con los perímetros que se establecieron anteriormente.

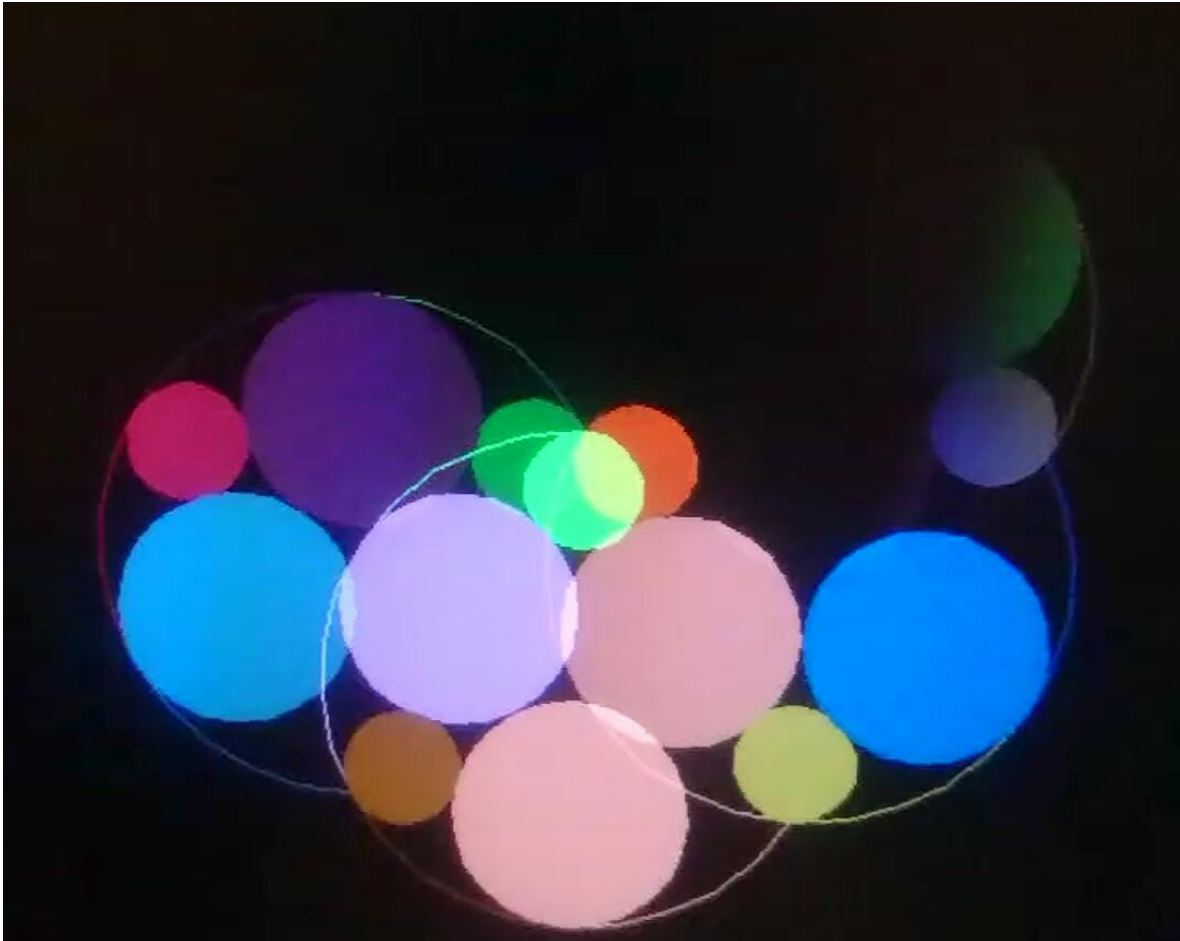
Si hubo algún error durante el procedimiento anterior, el programa va a decir que deben ser números enteros mayores a 0.

```
>>> ravel(t, 0.5)
```

Deben ser números enteros mayores que 0

Circo de Colores

La manera creativa de demostrar el Círculo de Ravel es de la siguiente manera:



Una vez que se da la orden, aparecen un conjunto de círculos de Ravel con colores diferentes y realizando diferentes secuencias.

Pasos

1. Se debe poner a correr el programa, esto se hace cuando ya se está en el documento de Fractales y se pulsa la tecla F5, esto hace que aparezca el shell de Python.
2. Para este programa, se debe poner en el primer renglón del shell de Python

```
>>> ravelMod(profundidad,radio)
```

3. Donde se menciona profundidad, se debe de colocar un número que defina la profundidad de la que desea que sea el fractal, que va a decir la cantidad de veces que se va a repetir la recursión. El radio define de que tamaño va a ser la figura inicial. Éste parámetro también debe ser un número entero y positivo. **Se recomienda `ravelMod(2,100)`**
4. A continuación se presiona RETURN o ENTER y esto va a hacer que se abra una nueva ventana, la de turtle, donde va a aparecer el dibujo del fractal con los perímetros que se establecieron anteriormente.

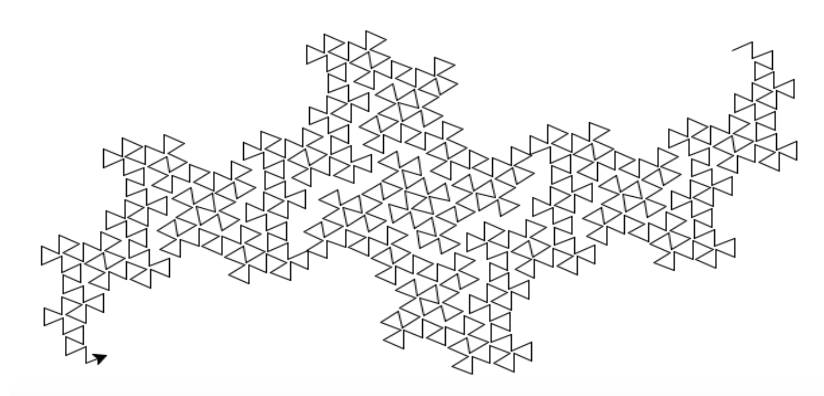
Si hubo algún error durante el procedimiento anterior, el programa va a decir que deben ser números enteros mayores a 0.

```
>>> ravelMod(t, 2.5)
```

Deben ser números enteros mayores que 0

Curva de Satie

El objetivo de este programa es lograr algo parecido al siguiente fractal:



Consiste en dibujar una línea recta, y esta se divide en tres líneas (profundidad 1). La línea del medio es la mitad del tamaño del espacio que hay entre el inicio y el final. Las otras líneas se definen por medio del teorema de Pitágoras. Cada línea se sustituye por el mismo patrón con la excepción de que es en zig zag.

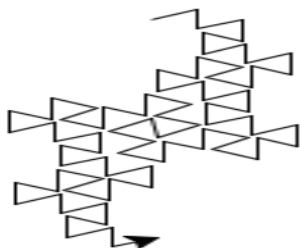
PROFUNDIDAD 1



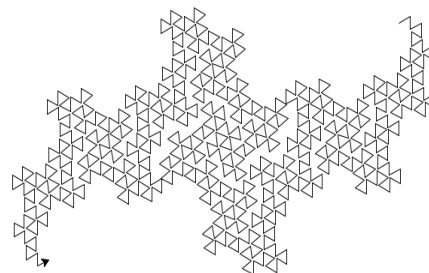
PROFUNDIDAD 2



PROFUNDIDAD 4



PROFUNDIDAD 6



Pasos

1. Se debe poner a correr el programa, esto se hace cuando ya se está en el documento de Fractales y se pulsa la tecla F5, esto hace que aparezca el shell de Python.
2. Para este programa, se debe poner en el primer renglón del shell de Python

```
>>> satie(profundidad,lado)
```

3. Donde se menciona profundidad, se debe de colocar un número que defina la profundidad de la que desea que sea el fractal, que va a decir la cantidad de veces que se va a repetir la recursión. El lado define de que tamaño va a ser la figura inicial. Éste parámetro también debe ser un número entero y positivo.
4. A continuación se presiona RETURN o ENTER y esto va a hacer que se abra una nueva ventana, la de turtle, donde va a aparecer el dibujo del fractal con los perímetros que se establecieron anteriormente.

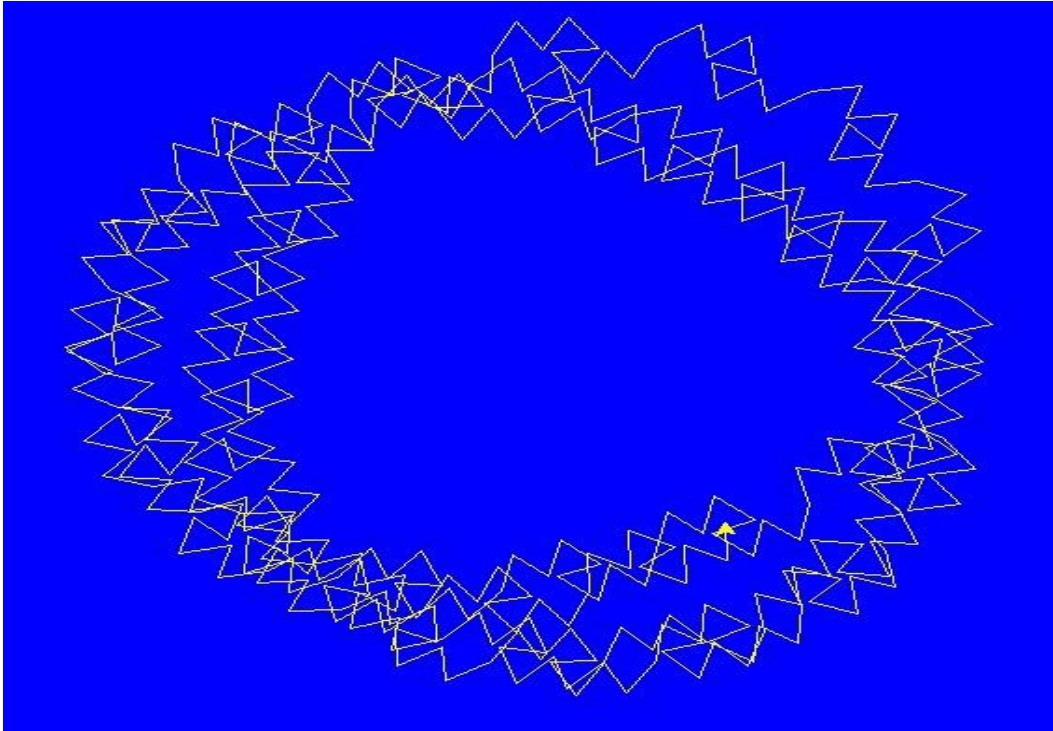
Si hubo algún error durante el procedimiento anterior, el programa va a decir que deben ser números enteros mayores a 0.

```
>>> satie(t, 0.5)
```

Deben ser números enteros mayores que 0

Espiral de Satie

La manera creativa de demostrar la Curva de Satie es de la siguiente manera:



Una vez que se da la orden, aparecen un conjunto de espirales, dependiendo de los parámetros que se le indique.

Pasos

1. Se debe poner a correr el programa, esto se hace cuando ya se está en el documento de Fractales y se pulsa la tecla F5, esto hace que aparezca el shell de Python.
2. Para este programa, se debe poner en el primer renglón del shell de Python

```
>>> satieMod(profundidad,lado)
```

3. Donde se menciona profundidad, se debe de colocar un número que defina la profundidad de la que desea que sea el fractal, que va a decir la cantidad de veces que se va a repetir la recursión. El lado define de que tamaño va a ser la figura inicial. Éste parámetro también debe ser un número entero y positivo. **Se recomienda `satieMod(2,100)` ; `satieMod(1,100)` ; `satieMod(2,50)`**
4. A continuación se presiona RETURN o ENTER y esto va a hacer que se abra una nueva ventana, la de turtle, donde va a aparecer el dibujo del fractal con los perímetros que se establecieron anteriormente.

Si hubo algún error durante el procedimiento anterior, el programa va a decir que deben ser números enteros mayores a 0.

```
>>> satieMod(t, 2.5)
```

Deben ser números enteros mayores que 0

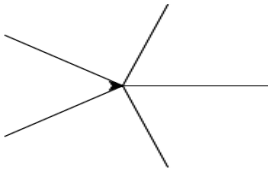
Estrella de Debussy

Con este programa se pueden crear varias diferentes figuras.

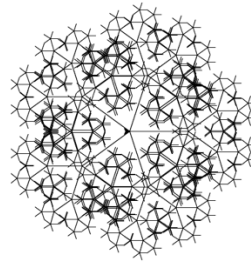
Hay cuatro parámetros diferentes por definir: profundidad, lado, cantidad de lados (n) y relación. Se crea con líneas, las cuales van a crear una estrella o un asterisco. La cantidad de picos en la estrella o el asterisco se elige con el parámetro de cantidad de lados o n. El dibujo original va a tener n cantidad de picos, la siguiente parte del dibujo va a ser (n-1). La relación va a definir en cuanto se divide en cada recursión. El lado va a definir el tamaño del fractal y la profundidad define cuantas veces se va a repetir el proceso.

Por ejemplo:

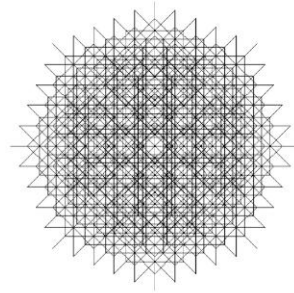
debussy(1, 100, 5, 0.5)



debussy(5,100,5,0.6)



debussy(5,50,8,1)



Pasos

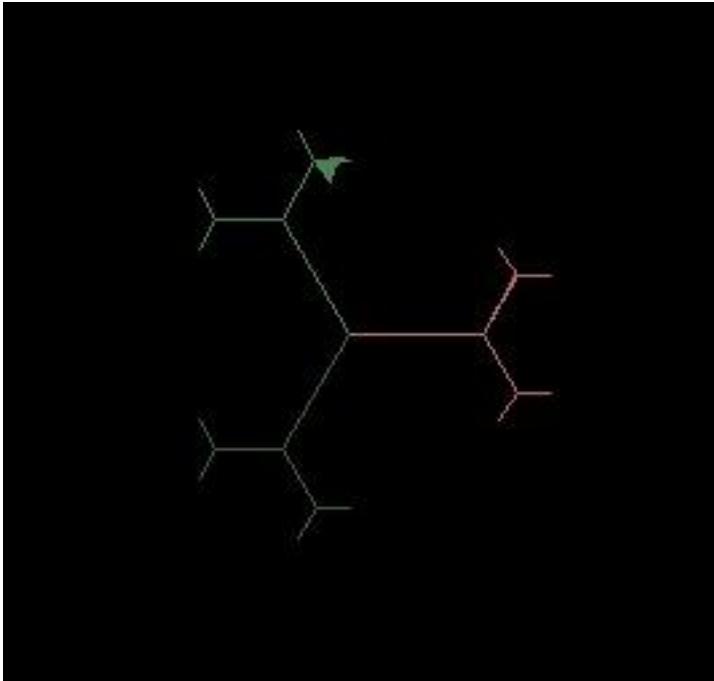
1. Se debe poner a correr el programa, esto se hace cuando ya se está en el documento de Fractales y se pulsa la tecla F5, esto hace que aparezca el shell de Python.
2. Para este programa, se debe poner en el primer renglón del shell de Python

```
>>> debussy(profundidad,lado,n,relacion)
```

3. A continuación se presiona RETURN o ENTER y esto va a hacer que se abra una nueva ventana, la de turtle, donde va a aparecer el dibujo del fractal con los perímetros que se establecieron anteriormente.

ColoredStar

La manera creativa de demostrar la Estrella de Debussy es de la siguiente manera:



Una vez que se da la orden, aparecen un conjunto de ramas, con diferentes colores, que aparecen aleatoriamente.

Pasos

1. Se debe poner a correr el programa, esto se hace cuando ya se está en el documento de Fractales y se pulsa la tecla F5, esto hace que aparezca el shell de Python.
2. Para este programa, se debe poner en el primer renglón del shell de Python

```
>>> debussyMod(profundidad,lado,n,relacion)
```

3. Se recomienda **debussyMod(3,50,3,0.5)**

4. A continuación se presiona RETURN o ENTER y esto va a hacer que se abra una nueva ventana, la de turtle, donde va a aparecer el dibujo del fractal con los perímetros que se establecieron anteriormente.

Si hubo algún error durante el procedimiento anterior, el programa va a decir que deben ser números enteros mayores a 0.

```
>>> debussyMod(t, 2.5,7,4)
```

Deben ser números enteros mayores que 0