ART INTEGRATED LEARNING

IN

COMPUTER SCIENCE

NAME: Abhinandan Saha

CLASS: XII, SEC: H, SESS:Afternoon

ROLL: 01, I.D.NO: 06 0031

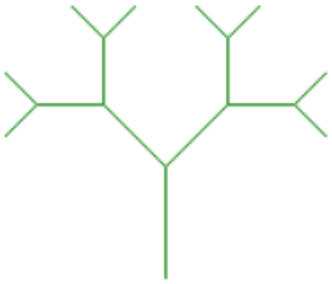
TOPIC: RECURSIVE FUNCTIONS AND THE MANDALA THANGKA ARTFORM OF LADAKH

The mandala, as a religious art in Tibetan Buddhism, thrives well in Ladakh.

Drawing a mandala pattern of thangka style is both time and effort consuming and requires mastery due to intricate details. However, an iterative approach can be used to model and generate mandala thangka patterns, namely the star, crescent, and lotus flower motifs using computer programs. Experimental results show that computer programs can efficiently generate beautifully-layered and colorful traditional mandala patterns. This significantly reduces the time and effort in manual production and help to the digitization of this great heritage.

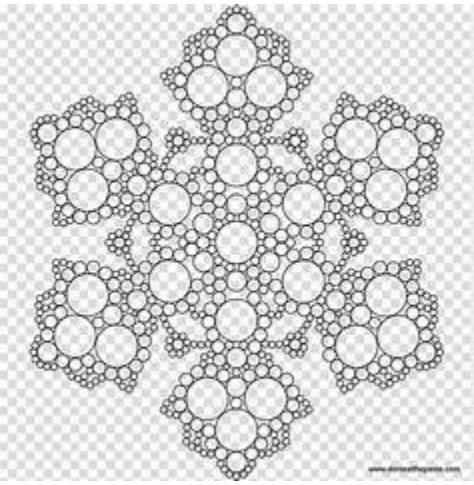
The Fractal Tree

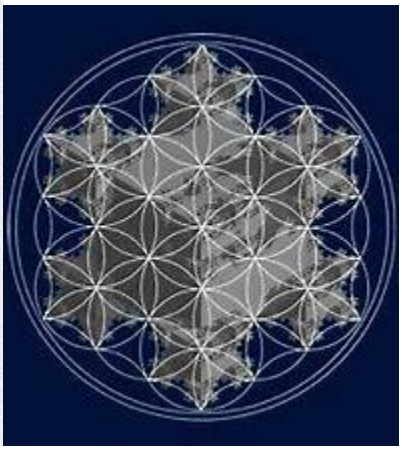
```
#ABHINANDAN SAHA, XII, H, AFTERNOON, 01, 060031
import turtle
MINIMUM_BRANCH_LENGTH = 10
def build_tree(t, branch_length, shorten_by, angle):
    if branch_length > MINIMUM_BRANCH_LENGTH:
        t.forward(branch_length)
        new_length = branch_length-shorten_by
        t.left(angle)
        build_tree(t, new_length, shorten_by,angle)
        t.right(angle*2)
        build_tree(t, new_length, shorten_by, angle)
        t.left(angle)
        t.backward(branch_length)
tree=turtle.Turtle()
tree.hideturtle()
tree.setheading(90)
tree.color('green')
build_tree(tree, 50, 10, 45)
turtle.mainloop()
```



The parameter values have been changed to 50, 10 and 45 for branch_length, shorten_by and angle respectively during function call in the main code, to give rise to a different pattern.

THE KOCH SNOWFLAKE





untitled0.py* untitled1.py*

```
#ABHINANDAN SAHA, XII, H, AFTERNOON, 01, 060031
import turtle
def koch_curve(t, iterations, length, shortening_factor, angle):
    if iterations==0:
        t.forward(length)
    else:
        iterations=iterations-1
        length=length/shortening_factor
        t.left(angle)
        koch_curve(t, iterations, length, shortening_factor, angle)
        t.right(angle*2)
        koch_curve(t, iterations, length, shortening_factor, angle)
        t.left(angle)
        koch_curve(t, iterations, length, shortening_factor, angle)
t=turtle.Turtle()
t.hideturtle()
for i in range(4):
    koch_curve(t, 5, 200, 3, 30)
    t.right(120)
turtle.mainloop()

```

Here the parameter values during function call in main code have been changed to 5, 30 for iterations and angle giving rise to a different form, due to the change in parameters and the reaching of base case at a different time and situation.