Project 3: Draw a Fractal Tree in C++

Programming Environment

1. There is no official library of drawing graphs in C++. The simplest way is to use graphics.h in Turbo C++ in online gdb.

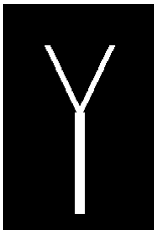
2. Functionality of graphics.h can be found at https://home.cs.colorado.edu/~main/bgi/doc/

   * There are other graphics libraries for C++, but please use graphics.h in Turbo C++, otherwise, I cannot test your code.

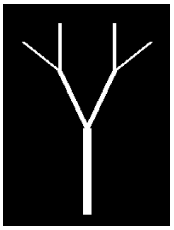
3. A sample program to draw three lines using graphics.h is shown in https://www.geeksforgeeks.org/draw-line-c-graphics/. To make the code work in online gdb with Turbo C++, modify line `initgraph(&gd, &gm, "");` to
   `initgraph(&gd, &gm, "/tc/bgi");`

4. Sample output of fractal tree
   a tree with one layer (trunk only)

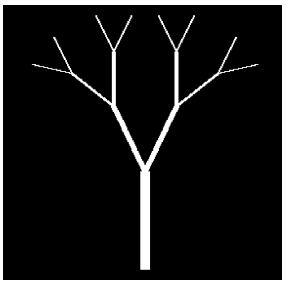
   a tree with two layers



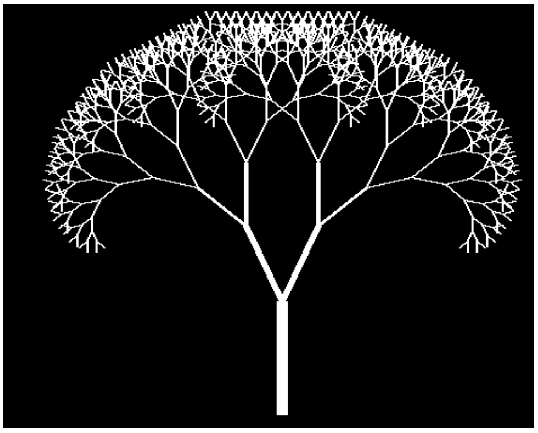
   a tree with three layers



a tree with four layers



A tree with 10 layers.



Parameters of a tree

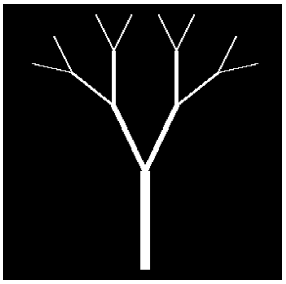
```
int x = 320; //x-coord of the root
int y = 386; //y-coord of the root
int length = 100; //length of the trunk of the tree
int width = 10; //width (thickness) of the trunk of the tree
double startAngle = M_PI / 2; //angle of the trunk starts at 90 degree,
                              //since the trunk is vertical to the ground.
double angle = M_PI / 7; //acute angle between a branch and
                         //the branch growing out from the first branch.
                         //or half of the angle of a v-shape branch.
int layer = 4; //number of layers of the tree

drawTree(x, y, length, width, startAngle, angle, layer);
```

To make the tree look more natural, when the tree moves up one layer, the length of the branch is ¾ of the previous one, and the width (thickness) of line representing a branch is 2/3 of the previous one.

We also set the minimum of length is 10 (pixels) and the thickness of branch is 2. You can modify the above parameters to see how the tree changes.

Explanation of length and width of branches in different layers



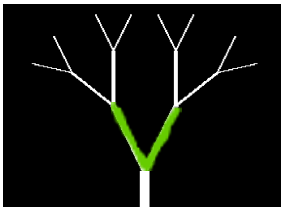
(1) Suppose the length of trunk is 100 and width is 10.
(2) The length of the 2nd layer branch is 100 * ¾ = 75 and the width is 10 * 2/3 = 6.
(3) In the third layer, the length is 75 * ¾ = 56 and the width is 6 * 2/3 = 4.
(4) In the branches of the fourth layer, the length is 56 * ¾ = 42 and the width is 4 * 2/3 = 2.
(5) In the branches of the fifth layer, the length is 42 * ¾ = 31 and the width is 2 * 2/3 = 1.
But we would like to set the minimum width to be 2, so the width is updated to be 2.

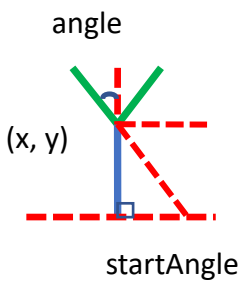
Assume that the length is not smaller than 10 and the width is not smaller than 2.

Key of the program

Write a recursive function to draw the v-shape branches.

void vShapeBranch(int x, int y, int length, int width, double startAngle, double angle, int layer)





For the left green branch, the start point's coordinate is (x, y), and the angle to the horizontal line is startAngle + angle, where angle is defined as the acute angle between the blue branch where the green branch extends from. So, the angle between the left green branch and the horizontal line is (startAngle + angle).

Given the length of left green branch, calculate the coordinate of the higher end of the left green branch using cos and sin function on the corresponding angle. Draw a line for the left green branch using line function from graphics.h.

Similarly, draw the right green branch.

Next, grow another v-shape branch from the higher end of the left green branch. Do similar thing for the higher end of the right green branch. Adjust corresponding values for x, y, startAngle, length, width, and layer.

Draw a tree by adding a trunk below the v-shape branches of given layers.

Function to draw a line with given width (aka thickness)

File graphics.h does not provide a function to draw a line with given thickness. The library has only two width, 1 pixel and 3 pixels, for a line. I write a function to draw a line from (x1, y1) to (x2, y2) with given width. You may use it or adapt it for your code.

```
//Draw a line from (x1, y1) to (x2, y2) with given width, where width is at least 2.
void drawLine(int x1, int y1, int x2, int y2, int width)
{
   for (int i = -width/2; i < width/2; i++)
        line(x1+i, y1, x2+i, y2);
        //as long as (x2 + i) - (x1 +i) does not change,
        //y1 and y2 does not need to change.
}
```