

# About Lab 5

Lab 5 is our Recursion lab. You will write 3 programs, all of which allow the user to run multiple recursive functions. Altogether there are 10 recursive functions to write. By the end of this lab you will be much more comfortable with the idea of recursion.

The first program, `recnum.py`, asks you to write 3 functions that recurse on numbers. In each case there is a description of the recursive algorithm.

The second program, `recstr.py`, asks you to write 3 functions that recurse on strings. Again there is a description of the recursive algorithm. Note that on the palindrome function `pals(s)`, you are asked to recurse directly on `s` and not call a function `reverse(s)`.

For this question you need to recurse on a string that is all of string `s` except for its first and last letters.

One way to get this is

```
s[1:-1]
```

An equivalent way is

```
s[1:len(s)-1]
```

More interesting are the recursive pictures from this lab.

In all of these pictures there is a recursive draw function. The arguments to this function are

- a) the canvas
- b) something that describes where to draw, usually the upper left corner and size of a rectangle
- c) a level

At the lowest level you simply draw a figure in the rectangle and quit.

At higher levels you may or may not draw something, then you recurse on subrectangles at the next lower level.

For example, the code for the bob figure looks like this:

```
def bob(x, y, size, level, canvas):
```

```
    if level > 0:
```

```
        canvas.drawSquareFill(x+size//4, y+size//4, size//2)
```

```
        bob(x, y, size//2, level-1, canvas)
```

```
        bob(x+size//2, y, size//2, level-1, canvas)
```

```
        bob(x, y+size//2, size//2, level-1, canvas)
```

```
        bob(x+size//2, y+size//2, size//2, level-1, canvas)
```

The Snowflake curve is a bit different. For this one you need some pen drawing routines from the Picture module. There is a virtual pen in the module. You can set it down at position (x, y) on the canvas with

```
canvas.setPosition(x, y)
```

The pen has an orientation, which is initially moving to the right. You can move it forward in this direction with

```
canvas.drawForward( distance )
```



You can change the pen's direction with  
`canvas.rotate(angle)`

Positive angles rotate clockwise, negative ones  
counter-clockwise.

Here is code for drawing a triangle. Be sure you understand this before you try to draw the snowflake.

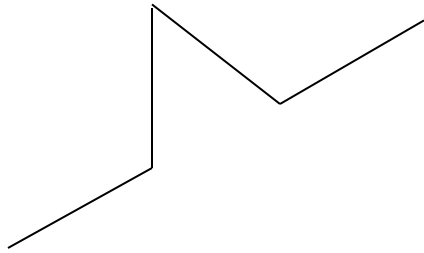
```
canvas = picture.Picture(400, 400)
canvas.setOutlineColor( 0, 0, 255 )
canvas.setPosition(100, 300)
canvas.rotate(-60)
canvas.drawForward(200)
canvas.rotate(120)
canvas.drawForward(200)
canvas.rotate(120)
canvas.drawForward(200)
```

For the snowflake, you draw a triangle just like the previous code, only you replace the `canvas.drawForward( )` calls with calls to a recursive `drawLine(size, level, canvas)` function. Note this carefully: the snowflake itself is not recursive; it is the `drawLine( )` function that is recursive.

The `drawLine( )` function at level 0 just calls `canvas.drawForward( )`.

At higher levels drawLine( ) makes 4 recursive calls to itself, replacing a line at one level with a bumpy line at the next lower level:

drawing



instead of

