Possible instructions

pu:	Pen Up – does not draw while moving
pd:	Pen Down – will draw when moving
fd(x):	Forward – move forward a specified distance
rt(x):	Right turn – turn right a specified # of degrees (no movement)
lt(x):	Left turn – turn left a specified # of degrees (no movement)
repeat(x):	Repeat the indented instructions x # of times
incr(x,i):	Increment variable *x* by amount *i*

Examples:

To move forward 1 unit:	**fd(1)**
To turn right 60 degrees:	**rt(60)**
To repeat the indented instructions 3 times:	**repeat(3)**
To increment *x* by 2:	**incr(x,2)**

1) What are the instructions required to draw a square with sides of length 2 in the lower right quadrant, starting from the cursor?
2) Those instructions can be used to create a new instruction called **Square**.
3) Using the new **Square** instruction, what sequence of instructions is required to draw the 4-square figure below?
4) Using the new **Square** instruction, what sequence of instructions is required to draw the concentric squares figure below – with sizes of 1, 2, and 3?
5) How would we use the **repeat** instruction to consolidate the repeated code in the **Square** function?
6) How would we use the **repeat** instruction to further modify the code in the 4-square drawing to remove the repeated code?
7) Use the **incr** instruction in the concentric squares problem to make the code for drawing each square repeatable.
8) Use **repeat** to consolidate the concentric squares problem.

Square(x) – draws a square CW (clockwise) with sides of length x

```
    pd
    fd(x)
    rt(90)
    fd(x)
    rt(90)
    fd(x)
    rt(90)
    fd(x)
    rt(90)
    pu
```

There is unnecessary repetition of code here, if we had an instruction that allowed us to repeat code a certain # of times, we could reduce the Square function to just a few lines:

Repeat(x) – repeats the indented code x # of times…

```
    Instruction 1
    Instruction 2
    Etc…
```

Use the Repeat instruction to consolidate the code in the Square function.

Solution: The Square function can be rewritten as:

```
Square(x)
    pd
    repeat(4)
        fd(x)
        rt(90)
    pu
```

Now, let's look at the code we wrote to draw the 4-square figure:

```
Square(2)

rt(90)

Square(2)

rt(90)

Square(2)

rt(90)

Square(2)

rt(90)
```

This repetition looks familiar – how can we use the **repeat** instruction to consolidate the instructions to draw the 4-square figure?

Solution: The code for the 4-square figure becomes:

```
repeat (4)
          Square(2)
          rt(90)
```

Now let’s look at the solution to the concentric squares problem:

Start at the center, move to the upper left corner, and then use Square to draw in clockwise fashion, returning to the same corner, in the same orientation. After we’re done return the curser to the center, with same orientation as it started with.

```
pu
fd(-1/2)
lt(90)
fd(1/2)
rt(90)
Square(1)
fd(-1/2)
lt(90)
fd(1/2)
rt(90)
Square(2)
fd(-1/2)
lt(90)
fd(1/2)
rt(90)
Square(3)
pu
fd(1 1/2)
rt(90)
fd(1 1/2)
lt(90)
```

This is almost repeated code. What is not the same? How can we handle it?

Assume we have a variable ***x*** which can store a value.

Introduce an increment instruction **incr(x,i)** which adds the value **i** to the variable **x**

Now modify the concentric squares problem to use **incr(x,i)** to set **x** for each square.

Solution: Here’s the concentric squares problem using **incr(x,i)**:

```
pu
x = 1
fd(-1/2)
lt(90)
fd(1/2)
rt(90)
Square(x)
incr(x,1)
fd(-1/2)
lt(90)
fd(1/2)
rt(90)
Square(x)
incr(x,1)
fd(-1/2)
lt(90)
fd(1/2)
rt(90)
Square(x)
incr(x,1)
pu
fd(1 1/2)
rt(90)
fd(1 1/2)
lt(90)
```

Now we have repeated code! (shown in red)

Rewrite the concentric squares problem using repeat(x) and incr(x,i) to remove the repeated code:

Solution: Here’s the updated code to draw the concentric squares:

```
pu
x = 1
repeat(3)
    fd(-1/2)
    lt(90)
    fd(1/2)
    rt(90)
    Square(x)
    incr(x,1)
pu
fd(1 1/2)
rt(90)
fd(1 1/2)
lt(90)
```

Once again, we have repeated code, this time for moving the pen to the upper-right corner of the next square:

How would you consolidate that code, passing in a value similar to the Square routine we created? Create a Diag instruction that moves the cursor to the required position, passing in the amount to move in the x and y directions. It retains the original direction, and does not change the pen setting.

Solution:

```
Diag(x)
    Pu
    Fd(-x)
    Lt(90)
    Fd(x)
    Rt(90)
```

We could change this version of Diag, so that we can also pass in the degrees – positive is CCW, negative is CW.

We can further add an indication that the pen should be up or down – now we can move diagonally directly at any angle and by any distance and choose to draw or not:

Diag(p, d, x) – p = dn/up, d = degrees (+ CCW, - CW), x = length of diagonal.

We need to add a conditional test – an if statement to determine whether we're moving (pu) or drawing(dn).

```
Diag(p, d, x)
    If p equal up
        Pu
    Else
        Pd
    Lt(d)
    Fd(x)
    Rt(d)
    pu
```

But really, we could extend this to do all of our drawing, we could substitute Diag(dn,90,x) for pd, lt90, fd(x)

Now we can draw the concentric squares with the following instructions:

```
Diag(1/2)

Square(1)

Diag(1/2)

Square(2)

Diag(1/2)

Square(3)

Diag(1/2)

Square(4)
```

This looks familiar – repeated code, but not quite the same, since the square instruction uses different lengths each time. Let’s introduce another instruction: incr(x, i), which increments x by the amount i. Let’s also initialize the variable x to 1:

```
X = 1

Repeat (4)

    Diag(1/2)

    Square(x)

    Incr(x,1)
```