

Introduction

You are encouraged to work in pairs

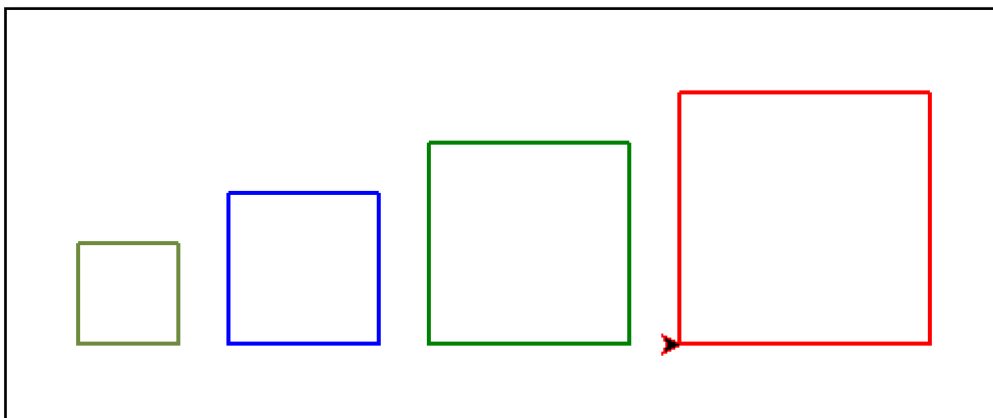
The aim of this exercise is to encourage familiarity with the process of problem solving through programming. You will be using the python Turtle library (code someone else has written) to make your job easier. Documentation for the Python Turtle library you will be using can be found [here](#).

There are a set of questions below, you can complete as many as you want - they are not graded. Create a new python program for each question. Make use of the cut, paste and edit operations in your editor as much as possible to save typing.

Question 1

Start off by running the sample python turtle program: `try_turtle.py`, see what it does. Now experiment to see what happens when you introduce minor mistakes into the code such as spelling mistakes or altering the indentation of lines.

`try_turtle.py` contains the code to draw a square. Copy the file to a new file named `question_1.py`. Add code to draw a series of squares so that the output looks like the image below. Send your turtle back to the start of the shape each time. After drawing the shape your turtle should be facing the direction it started from too.



Question 2

Copy your program file and give it a new name: `question_2.py`. You have probably noticed that there is a lot of repeated code in your last program. Rewrite the code for each square to use a **for loop** for each square. Here is an example of the code that will draw the first square in the picture above:

```
# Draw the smallest square

for counter in range(4):
    forward(50)
    left(90)
```

You can see that the commands to draw each side are repeated 4 times. Try it out and then amend the rest of your program to use for loops.

Question 3

Use the incomplete program `question_3.py`. This python program contains the function `draw_square`. A function gives us another way of reusing code, see below. You will see that the function name is preceded by the key word `def` and that we can provide the function with a value (called a parameter), called `side_length`.

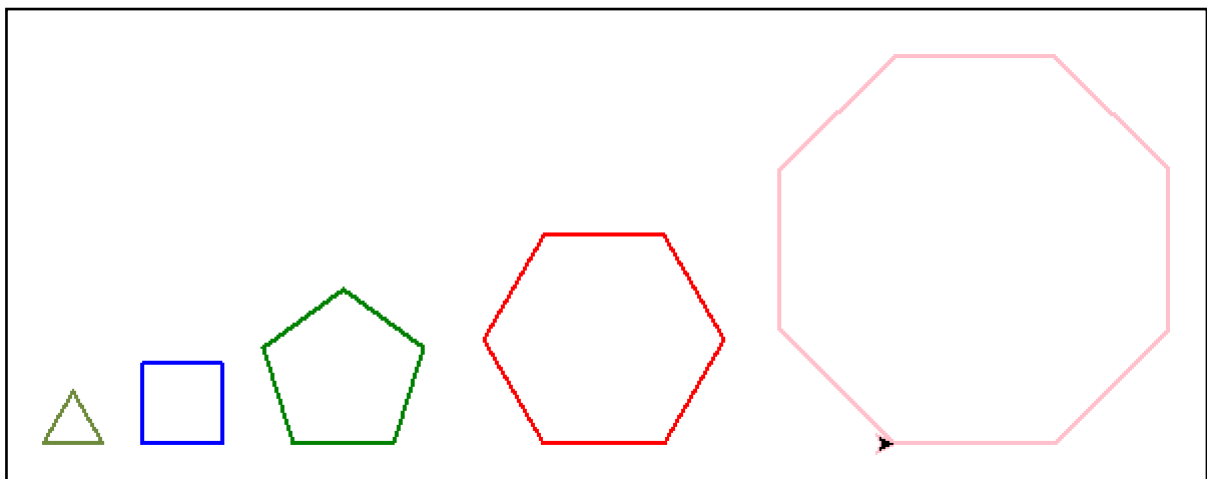
```
line 1  #
line 2  # This is a function to draw a square
line 3  #
line 4  def draw_square(side_length):
line 5      for counter in range(4):
line 6          fd(side_length)
line 7          lt(90)
line 8
line 9  # This is how you use the function
line 10 draw_square(50)
```

Run `question_3.py` and see how it works. Now edit `question_3.py` so that it draws the four squares as before.

Question 4

Create a new program called `question_4.py`. Add the `draw_square` function from the last question and check that it works. Add new functions to your program to draw a triangle, pentagon, hexagon and octagon respectively: `draw_triangle`, `draw_square`, `draw_pentagon`, `draw_hexagon`, `draw_octagon`.

The size of the gaps between the shapes and the colours of the lines is not important.

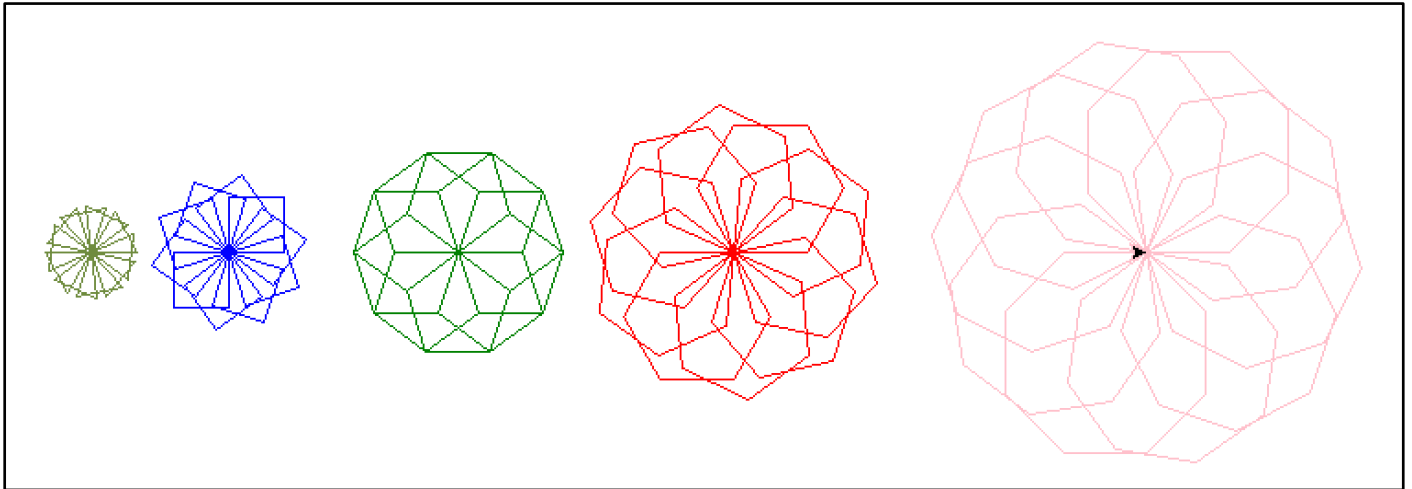


Question 5

If you know the number of sides that a polygon has, what arithmetic expression could you use to calculate the angle through which the turtle should turn? Use this arithmetic expression in your program to write a new program with a function: `drawNgon(sides, sideLength)`. Draw the shapes above with the new function.

Question 6

Write a program to produce this pattern through rotation:

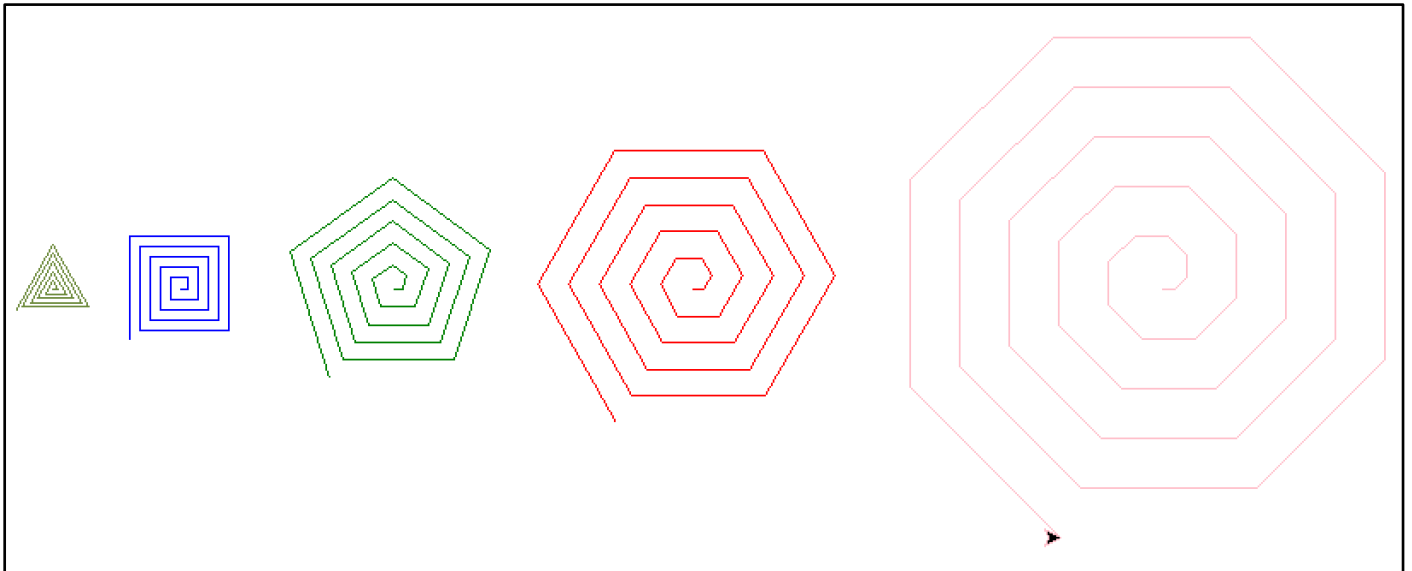


Note: the first shape is a rotated triangle, the second shape is a rotated square, and so on.

Hint: Use the functions you wrote in the last program and call the functions several times using a **for** loop.

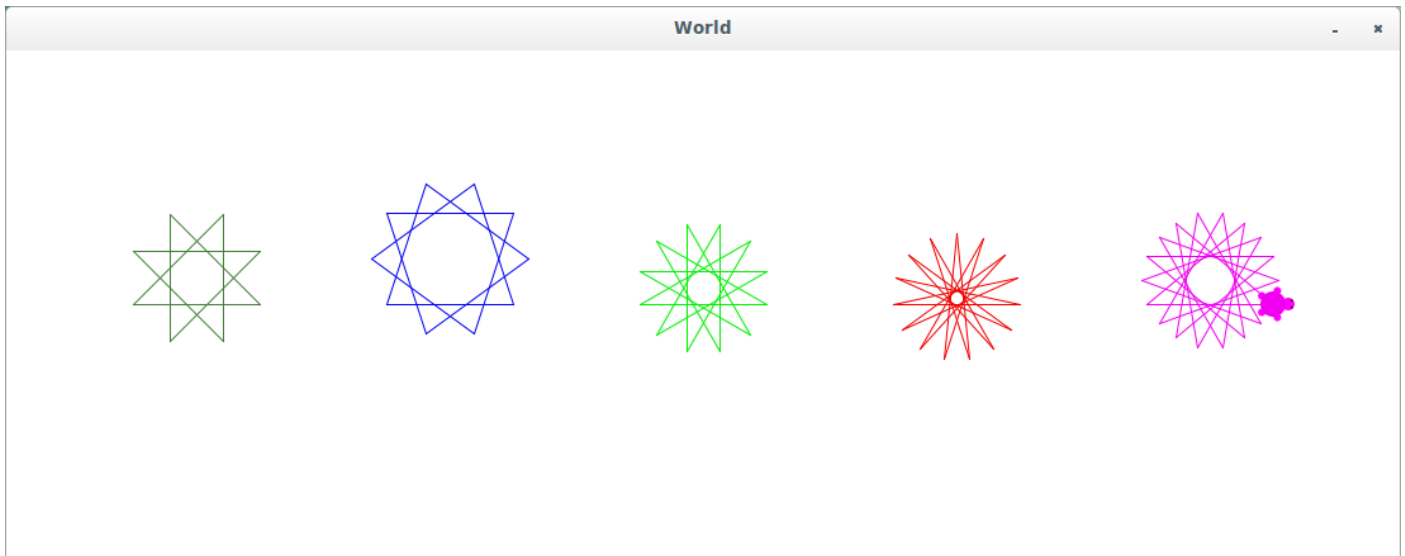
Question 7

Write a program to print spirals like this:



Question 8

This question is optional. Write a program to draw the series of stars shown in the image below. The stars have: 8, 10, 12, 15 and 18 points respectively:



Question 9

This question is optional. Create a design with the Turtle library, it can be anything you like, inspired by: Islamic art, fractals, tessellations.

Like to go further?

Investigate [fractals](#). Can you draw a pattern with Koch curves?