

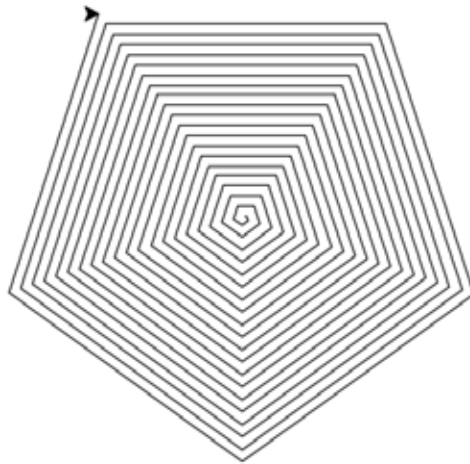
Midterm Practice Coding Problems

On the Spring 2022 Midterm for CS 110, you will be asked to write two Python programs: one easy and one hard. Below are some examples of the kinds of programming questions you will encounter on the Midterm.

Problem 1: Easy Turtle Program

Spiral Pentagon: Using the provided file **spiral_pentagon.py**, write a program that uses turtle graphics to draw a pentagon (5-sided) spiral that looks like the figure below. Each side is 2 pixels longer than the previous side, and the figure has 100 **sides** (not 100 hexagons). The program should be written as two functions:

- **main()**: creates the screen and the turtle, then calls **draw_spiral()**, passing the turtle as an argument, then waits for the user to click the screen before closing it.
- **draw_spiral(turtle)**: takes a turtle as a parameter and draws the figure.



Problem 2: Easy Program

Circle Perimeter: Starting with the starter file **circle_perim.py**, write a program that asks a user for two points (x_1, y_1) and (x_2, y_2) , then computes the perimeter of the circle for which (x_1, y_1) is the center and (x_2, y_2) is on the circle's perimeter. The program should define and use 3 functions:

- **distance(x1, y1, x2, y2):** computes and returns the distance between two points
- **perim(r):** computes and returns the perimeter of a circle with radius r
- **main():** asks user for input, calls the other functions, and prints the answer.

The input and output of the program should look similar to this (with different points):

```
Enter X for center point: 0
Enter Y for center point: 0
Enter X for perimeter point: 1
Enter Y for perimeter point: 1
Perimeter of circle centered at (0.0, 0.0) with radius
1.4142135 is 8.885765876
```

Hint1: Perimeter of a circle = $2\pi r$

Hint2: Distance between 2 points = $\text{sqrt}((x_2 - x_1)^2 + (y_2 - y_1)^2)$

Hint3: You might need to import a module

Problem 3: Hard Program

Password Test: For security reasons, computer passwords usually must meet certain requirements. Using the provided starter file `password_test.py`, write a program that checks if passwords entered by the user meet these security requirements:

- at least 9 characters
- no spaces or tabs
- at least 1 upper-case letter
- at least 1 lower-case letter
- at least one digit
- at least one special character (not letter, digit, space, or tab)

The program has two functions: `test_pwd(p)` and `main()`.

`test_pwd()` takes a password as a parameter, tests it for the above requirements, and returns **True** if the password is valid, **False** if not. It can use any of Python's built-in string functions and methods. It should **not** print anything or ask the user for anything.

`main()` displays the requirements for passwords, then asks the user to enter a password. After the user enters a password, `main()` calls `test_pwd()` to check if the password is valid. If `test_pwd()` returns **True**, print "Password OK." If `test_pwd()` returns **False**, print "Sorry, that is not a valid password." Count the number of valid passwords entered and the total number the user tried. Regardless of whether the password is valid or invalid, ask the user if they want to try another password (y or n). Validate user's response: if what user entered is not **y** or **n**, ask again for y or n. If the user enters **y**, loop back for another password. If the user enters **n**, the program prints "Your score: N valid passwords in M attempts." and quits.

The program's output should look like this (but with different user-entered passwords):

```
Password must have:
• at least 9 characters
• no spaces or tabs
• at least 1 upper-case letter
• at least 1 lower-case letter
• at least 1 digit
• at least 1 special character (not letter, digit, space,
or tab
Enter password: xyzzy
Sorry, that is not a valid password.
Want to try another password? (y or n): 9
Please enter y or n.
Want to try another password? (y or n): y
Enter password: xyzzyXYZZY@3
Password OK.
Want to try another password? (y or n): n
Your score: 1 valid passwords in 2 tries.
```