**Assignment #3**

For this assignment you will be writing 3 programs which should be saved independently as their own “.py” file. The filename you should use for each program is outlined in the sections below. When you’re finished you should submit your programs to the Assignment #3 category inside of NYU Classes.

You may work by yourself or with a partner. Remember, collaborating just entails discussing approaches to the problems; you should not be writing actual code together. If you choose to work with someone else please be sure to give them credit in your source code and on NYU classes when you submit your work. Each member of a team must submit their own unique copy of the assignment to get credit for the assignment.

[Problem 1 |](https://emilydidthis.github.io/CSCI-UA.0002-Fall22/assignments/03.html#problem-1-valid-triangle-tester)[Problem 2 |](https://emilydidthis.github.io/CSCI-UA.0002-Fall22/assignments/03.html#problem-2-guess-the-number-game-with-3-attempts)[Problem 3](https://emilydidthis.github.io/CSCI-UA.0002-Fall22/assignments/03.html#problem-3-valid-date-checker)

Problem #1: Valid Triangle Tester

* Prompt the user to enter in 3 points on a standard Cartesian coordinate plane. You can assume the user will enter these values as floating point numbers.
* Compute the distance between each points - you will be computing 3 distance values. Use the distance formula to do this: ( (x1-x2)2 + (y1-y2)2 ) 0.5 … note that you can raise a value to the 0.5 power to compute the square root, or you can use the sqrt() function in the math module - your choice!
* Report the distance to the user of each side, rounded to 2 decimal places
* Next, determine if the three points could form a valid triangle. You can assume that the triangle is valid by checking the following. **Note that ALL THREE of these conditions must be met in order for a triangle to be valid)**:

(a) Side 1 + Side 2 must be longer than Side 3

(b) Side 2 + Side 3 must be longer than Side 1

(c) Side 3 + Side 1 must be longer than Side 2

* If the triangle is valid, report whether it is an equilateral, isosceles, or scalene triangle:
  + Equilateral Triangle: all sides of the triangle have the same length
  + Isosceles Triangle: only two sides of the triangle have the same length
  + Scalene Triangle: all sides of the triangle have different lengths

Hint: when comparing the size of each triangle it may be helpful to compare a “rounded” or “formatted” version of the length of each side. Due to floating point inaccuracies you may run into a where two values are virtually identical but Python will evaluate them to be different (i.e. 0.000000001 != 0.0)

**Extra Credit**: Also report if the triangle is a right triangle (i.e. it has one angle of 90 degrees). You can do this by using the Pythagorean Theorem on the sides of the triangle - a2 + b2 = c2. Hint: “c” (the hypotenuse) MUST be the largest side for the Pythagorean Theorem to work. Use an “if” statement to find the largest side and then test to see if the theorem holds. Hint #2: round your numbers to 0 decimal points for the purpose of this calculation. For example, round(a2) + round(b2) == round(c2).

Here are some sample runnings of the program, as well as a diagram that will help you to visualize the problem:

图表, 折线图

描述已自动生成

Valid and Invalid Triangles

**Run #1**

Valid Triangle Tester

Enter Point #1 - x position: 0

Enter Point #1 - y position: 0

Enter Point #2 - x position: 50

Enter Point #2 - y position: 86.6

Enter Point #3 - x position: 100

Enter Point #3 - y position: 0

The length of each side of your test shape is:

Side 1: 100.00

Side 2: 100.00

Side 3: 100.00

This is a valid triangle!

This is an equilateral triangle

**Run #2**

Valid Triangle Tester

Enter Point #1 - x position: 0

Enter Point #1 - y position: 0

Enter Point #2 - x position: 0

Enter Point #2 - y position: 100

Enter Point #3 - x position: 100

Enter Point #3 - y position: 0

The length of each side of your test shape is:

Side 1: 100.00

Side 2: 100.00

Side 3: 141.42

This is a valid triangle!

This is an isosceles triangle.

This is also a right triangle.

**Run #3**

Valid Triangle Tester

Enter Point #1 - x position: 110

Enter Point #1 - y position: 100

Enter Point #2 - x position: 75

Enter Point #2 - y position: 25

Enter Point #3 - x position: 0

Enter Point #3 - y position: 0

The length of each side of your test shape is:

Side 1: 82.76

Side 2: 148.66

Side 3: 79.06

This is a valid triangle!

This is a scalene triangle

**Run #4**

Valid Triangle Tester

Enter Point #1 - x position: 0

Enter Point #1 - y position: 0

Enter Point #2 - x position: 50

Enter Point #2 - y position: 50

Enter Point #3 - x position: 100

Enter Point #3 - y position: 100

The length of each side of your test shape is:

Side 1: 70.71

Side 2: 141.42

Side 3: 70.71

This is not a valid triangle

Comment your source code and describe your code to someone who may be viewing it for the first time. Also include your name, the date, your class section and the name of your program at the top of your file.

Problem #2: Guess the Number Game with 3 Attempts

Write a program that lets the user guess an integer between 1 and 10. The integer should be randomly selected by your program when it starts up. Your player is allowed to guess three times - if they guess correctly, they win and the program ends. If they guess incorrectly they should be told if the number they guessed was too high or too low and then be asked to guess another time. At the end of the program you should display the secret number as well as how many attempts it took to guess that number.

Note: This program does not require a “while” loop – if you have gone ahead and looked at Module #4 that’s fantastic, but you are not allowed to write this program using this technique. Also, you are not allowed to use the “sys.exit()” function (or any other similar function) to cause your program to prematurely quit. Everything being asked of you can be done using “if”, “elif” and “else” statements.

Here are a few sample runnings of your program:

**Run #1**

I'm thinking of a number between 1 and 10!

What's your guess? 5

You got it!

The secret number was 5.

It took you 1 try to guess the number.

**Run #2**

I'm thinking of a number between 1 and 10!

What's your guess? 5

Too low, try again.

What's your guess? 8

Too high try again.

What's your guess? 7

You got it!

The secret number was 7.

It took you 3 tries to guess the number.

**Run #3**

I'm thinking of a number between 1 and 10!

What's your guess? 5

Too high, try again.

What's your guess? 1

Too low, try again.

What's your guess? 3

The secret number was 4.

You didn't guess the number.

Problem #3: Valid Date Checker

For this assignment you will be writing a “diagnostic” program that will be able to tell a user whether a date is valid. The program should begin by asking the user to supply a date in the following format: YYYYMMDD For example, 20200218 would be interpreted as “February 18th, 2020”. You can always assume that the user will supply an 8 digit integer. Next, the program should “diagnose” the date as one of the following:

* Whether the year provided is a Leap Year or not. During a Leap Year February has 29 days, and in non-Leap Years February only has 28 days. You can compute whether a year is a leap year by using the following algorithm:
  + A Leap Year is any year that can be exactly divided by 4 (such as 2016, 2020, 2024, etc)
    - … EXCEPT if it can be exactly divided by 100, then it isn’t (such as 2100, 2200, etc)
      * … EXCEPT if it can be exactly divided by 400, then it is (such as 2000, 2400)
* Note that the year will ALWAYS be valid. For example, 0000 is valid year, as is 9999
* Next, determine if the month / day combination is valid. For example, September 30th is valid, but September 37th is not. The following chart may help you to identify whether these combinations are valid or not:
  + Months with 30 days: September, April, June, November
  + Months with 31 days: January, March, May, July, August, October, December
  + Months with 28 days (29 during a leap year): February If the combination is invalid you should report this to the user.
* If the combination is valid you should report it to the user using the full name of the month. For example, if the date entered was 20210213 you would classify the month as “February”
* You should also report the date using the correct English label for the day. For example, “February 1st” - note the “st” after the number 1. In English the following are valid labels for the numbers 1-31:
  + 1, 21 and 31: st
  + 2 and 22: nd
  + 3 and 23: rd
  + All other numbers: th

Here are a few sample runs of the program:

Enter a date (YYYYMMDD): 20221320

2022 is NOT a leap year

This is not a valid date in 2022

-----------------------------------

Enter a date (YYYYMMDD): 20220931

2022 is NOT a leap year

This is not a valid date in 2022

-----------------------------------

Enter a date (YYYYMMDD): 20220930

2022 is NOT a leap year

September 30th 2022 is a valid date

-----------------------------------

Enter a date (YYYYMMDD): 20200229

2020 is a leap year

February 29th 2020 is a valid date

-----------------------------------

Enter a date (YYYYMMDD): 19950203

1995 is NOT a leap year.

February 3rd 1995 is a valid date.

-----------------------------------

Enter a date (YYYYMMDD): 20001021

2000 is a leap year.

October 21st 2000 is a valid date.