LAB #8
Lists/Arrays in Python

Check your grades on Canvas to make sure they are accurate!!! If not, then contact the TAs or myself!!!! Remember, there is no lab during Thanksgiving Holidays.

You need to use the TAs office hours to get extra help in understanding the material and what is required from an assignment or lab!!!

At this time, you can pair with someone in the lab or finish it on your own.

(5 pts) Practice Lists in Python

Let’s make a list, which is mutable. If you need to, review last Wednesday and this Monday’s slides.

Design and implement a program that generates a random number 1-20 and has a user guess the number. The user gets 5 guesses, and the numbers that the user guesses gets stored in an array. The user should not be able to guess the same number twice. If they do guess the same number, it should not count as a guess. You will need to iterate through the array with stored guesses to see if the user already guessed a number. The program should end when the user guesses the right number or uses all 5 guesses.

To generate a random number, import random. You will also need to seed the random number generator by making only one call to random.seed(). Read the documentation to figure out which random function you want to use to generate a random number, randint() or randrange(). https://docs.python.org/3/library/random.html

- You need to have a main function that serves as the starting place for the program. In this main you can call random.seed() to this only once!
- You need to have a function called get_guesses() and one called check_guesses(). Your get_guesses() will need to call your check_guesses() to make sure the guess is valid and can be stored in the array.
- Print the random number generated and the array of guesses in your main function.

(5 pts) Let’s practice 2-d lists/arrays

Write a program that creates a 2-d list/array corresponding to a multiplication table, i.e. the multiple of the row and column is stored at that location.

0 0 0
0 1 2
0 2 4

- You need to have a main function that serves as the starting place for the program.
You need to have a function called `make_multiplication_table()` and one called `print_table()`, which generates a pretty table like the one above. Your main function should call both these functions.

Make sure you get checked off by a TA prior to leaving the lab!!!!