Homework 6
Would you like to play a game?
Due October 29, 2021 at 5pm

In this homework, you will be doing a series of exercises designed to make you practice using lists, as well as to continue practicing using loops and functions. Each program you write **must** use at least one function (more if appropriate)! Each program should be in a separate Python file. You should use the elements of good basic Python style discussed in class.

Should you stumble across implementations of these programs out on the Internet, do not look at them. To do so would be a violation of the course collaboration policy. You may follow the links listed in the descriptions below, but do not Google around.

Learning Goals

1. Use lists to solve problems.
2. Gain additional practice with functions and loops.
3. Identify and fix errors.
4. Use elements of good basic coding style.

The Assignment

Write two programs to do the following two tasks. There are only two (rather than the three or four of recent homeworks) because we expect each program to be a bit longer than previous programs. You may also notice that the specifications for each program are not as detailed as previous programs. For instance, there is no sample output. You have greater freedom to choose the design of your approach and the appearance of your output. Feel free to ask course staff about the choices you are making.

1. **hw6a.py** Battleship (https://en.wikipedia.org/wiki/Battleship_(game)) is a classic two-player strategy guessing game**.** In the classic 1967 Milton Bradley version, each player positions five plastic ships on a ten-by-ten game board. Each ship is placed horizontally or vertically (not diagonally) covering a set number of squares. The five ships have sizes 5, 4, 3, 3, and 2 squares. Players then take turns guessing the location of their opponent's ships until them have found them all.

   For this assignment, you will only do the board generation part. Write a program to generate and display a Battleship-style ten-by-ten board containing five ships of the sizes above. Ships cannot overlap one another and cannot extend off the edge of the ten-by-ten board. Place each ship, one at a time, at a random position and orientation, covering as many squares as the size of that ship requires. If that placement violates the overlap or edge properties, try a different random position. Display the game board and the ships it contains before you begin placing ships and after each ship is placed.

Your solution should use lists to store the game board. You may display your board using simple characters or something more complicated, but it should be recognizable as a ten-by-ten board where some positions are occupied by ship portions (and others are empty). Particularly attractive displays may earn extra credit points.

2. **hw6b.py** Extend your implementation of a single Pig turn-to-20 from HW5 to simulate multiple runs of the turn and report statistics of the results. Your program should prompt the user for how many turns to simulate, store each simulated turn in a list of rolls, and then report out simulation statistics. Your statistics should include the average score, average run length (ie, number of rolls), and the sequence of rolls that make up the shortest and longest runs.

   Don't worry if you did not get your program from HW5 working correctly. We will go over a solution in class on Wednesday. Even if your single turn code is not working, you can start working on this program even before we talk about it in class – just focus on the other elements of the program.

How to turn in your homework
Turn in each program in its own file. When turning in your own assignment make sure to add your last name to the file name (for example: Rheingans_hw6a.py).