February 3rd 2018 Assignment (Accelerated):

<Main assignment>

Pig Latin is a game of alterations played on words. To make the Pig Latin form of an English word the initial consonant sound is transposed to the end of the word and an "ay" is affixed. Specifically, there are two rules:
(a) If a word begins with a vowel, append "yay" to the end of the word.
(b) If a word begins with a consonant, remove all the consonants from the beginning up to the first vowel and append them to the end of the word. Finally, append "ay" to the end of the word.

For example:
dog⇒ogday
scratch⇒atchscray
is⇒isyay
apple⇒appleyay

Write a program that repeatedly prompts for an English word to translate into Pig Latin and prints the translated word. If the user enters a period, halt the program.

Hints that you may find helpful: Find out how slicing strings works, Maybe the in operator will help

<If you have time…> Simulate Mastermind!

(See http://www.wikihow.com/Play-Mastermind for the actual board game for those who don't know how to play)

How to Play Mastermind (revised version for our purposes)

- The computer (codemaker) uses code pegs of six different "colors" (ABCDEF) to create a 4-color code. For example, BCFD or ADCB. (Assume colors cannot repeat. So AABB would be illegal)
- The goal of the game is for the codebreaker (the user) to correctly determine both the four colors selected and their position in the code.
- The codebreaker tries to guess the pattern, in both order and color, within 12 turns.
- Each guess is made by typing in the console.
- Then, the codemaker provides feedback by printing out 0 to 4 scoring pegs
  - A black scoring peg is placed for each code peg that is correct in both color and position.
  - A white peg is placed for a correct color peg placed in the wrong position.
  - The order of black/white pegs do not matter. For example, the codemaker can just print out "1B, 1W" or "1W, 1B" meaning the codebreaker received one black and one white peg for their recent guess.
  - Once feedback is provided, another guess is made.
  - This continues until either the codebreaker guesses correctly or 12 incorrect guesses are made. At this point, the codemaker should either declare a win or loss for the codebreaker.
  - Assume the codebreaker is not dumb and never puts in a 3 or 5 color code instead of 4.
  - The codemaker should always print the complete "board" state so the codebreaker can see the entire history of guesses made.
  - Example of a "board" after two tries from the user

1. input ABCD: 1B, 1W
2. input ADEF: 1B, 3W

Suggested step by step process

1. First start out creating and testing your code for comparing two 4-color codes and then correctly returning black and white pegs. That is the crucial part of your code.
2. Then, hardcode a codemaker's code into the program and work on having the codebreaker input the guess 12 times or until he/she gets it right.
3. Finally, work on randomly generating the codemaker's code.

If you have time, you can make it even more similar to the actual game by allowing colors to repeat in the codemaker's code.