HW1 is due on Sep 28, at 1pm (right before Thursday python class).

1. Use Python to take an input word from the user and determine if the word has a vowel (A, E, I, O, U, a, e, i, o, u) in it. Print a message to the user telling them if their word has a vowel in it or not. Extra goals: Count the number of vowels in the word. Find the index of the first vowel. Find the indexes of all vowels.
2. Use Python to take an input from the user and determine if it is divisible by 7. Print a message to the user telling them if their number is divisible by 7. Extra goal: write a script that checks if a number is divisible by 7 using Chika's test: take the last digit off a number, multiply it by 5 and add it to the remaining number, and repeat until the number is <10 or 49. If the number is 7 or 49, it is divisible by 7. Eg:

Take the number 2996:  
299 + 6x5 = 329 (this number is greater than 9 and isn't 7 or 49, so repeat)  
32 + 9x5 = 77  
7 + 7x5 = 42  
4 + 2x5 = 14  
1 + 4x5 = 21  
2 + 1x5 = 7 (this number is 7 or 49, so we are done)  
Therefore 2996 is divisible by 7.

1. The Fibonacci sequence is generated by adding the previous two terms, starting with 0 and 1. The first six elements are 0, 1, 1, 2, 3, 5. What is the greatest element of the Fibonacci sequence that is less than 268 billion (268,000,000,000), and what is its index, knowing that 0 is at index 0 and 1 is at index 1? Extra goals: Find the sum of even numbered Fibonacci numbers that are less than 4 million.
2. Bonus Question. Use Python to take an input from the user and find the greatest prime factor of the number the user input. Find all the prime factors of the input number.

Please read below instructions carefully, failing to follow basic requirements may lead to deduction of points.

1. Submit your solutions in one jupyter notebook. You can create your own or use the provided template.
2. Comment your code when necessary. You do not need to comment every line, but if your algorithm that is not straightforward, commenting will help TAs understand your logic.
3. Your solutions should be run without syntax error. It is OK if you cannot figure out a complete solution, write your algorithm in steps as comment instead.
4. Students can work individually, discuss in a group or refer to other resources, etc., but your submission should be your own work. If you copy someone else's work in whole or part, please **(a)** provide sufficient comments in the code to show that you understand how it works and **(b)** a reference to who or where you copied the code (e.g. a link to the source).
5. If you need additional time for any reason please reach out to both Piper and your TA to discuss an extension before the due date.
6. To increase your exposure to solutions to the homework, assignments will be de-identified and graded within TA groups (we will provide a grading rubric for you to use). If grades differ by more than 1 point, a TA or Piper will resolve the discrepancy.