THE UNIVERSITY OF ARIZONA.
DEPARTMENT OF COMPUTER SCIENCE

CSc 120: Fall 2019

Assignment 01: Part 1

Start: Tue Aug 27, 2019
Due: 8:00 PM, Wed Aug 28 2019

Important: Follow the directions given *exactly*: your code will be graded by software, so any deviation from the program spec can result in a significant loss of credit. If you are unsure about something, ask for clarification in office hours or Piazza (however, see the class's academic integrity policy about sharing code with others or using code written by others).

Please pay attention to the programming style guidelines for this class. For this assignment you will be notified of style violations but not penalized for them; style violations will be penalized in subsequent assignments.

Note
There are two problems in this assignment, which together require that you submit two files: **classify.py** (problem 1) and **median.py** (problem 2). Because of the way the auto-grader script works, in order to get full credit for your work you have to submit ***both*** these files each time you want to resubmit a solution to either problem.

Problem 1.
Write a program, in a file named **classify.py**, that behaves as follows:

1. It reads in an input string using the Python statement

   ```
   input("input: ")
   ```

   You can assume that the input string will have length at least 1.

2. It generates output as follows:
   - if the first character of the input string is a vowel (upper or lower case), the output is: **vowel**
   - if the first character is a consonant (upper or lower case), the output is: **consonant**
   - if the first character is neither a vowel nor a consonant, the output is: **neither**

Examples:

Input	Output
`apple`	`vowel`
`CSc 120`	`consonant`
`32F = 0C`	`neither`
`'Twas brillig`	`neither`

Problem 2.

Write a program, in a file named **median.py**, that behaves as follows:

1. It reads in an input string using the Python statement

   ```
   input("input: ")
   ```

   You can assume that the input string will have length at least 1.

2. It generates output as follows:
   - if the length of the input string is odd, the middle character of the string is printed;
   - if the length of the input string is even, the two middle characters of the string are printed.

Examples:

Input	Output
a	a
ab	ab
567	6
!@#$%&	#$