

Programming Project #3

Assignment Overview

This assignment involves coding and testing of a program that uses Python strings.

This assignment is worth 20 points (2% of course grade), and must be completed before 11:59 PM on Monday, September 25th.

Program Specifications

Your program will repeatedly prompt for words, collecting vowels and consonants, until you have collected all possible vowels (a, e, i, o, u) or at least five different consonants, whichever comes first. At that point the vowels, consonants and their counts are printed. One wrinkle is that the only consonants collected are those after the final vowel in a word.

Assume that a word that is input contains at least one vowel and is letters only (e.g., no spaces or hyphens).

Deliverables

The deliverable for this assignment is the following file:

`proj03.py` -- your source code solution

Be sure to use the specified file name and to submit it for grading via Mimir before the project deadline.

Assignment Notes:

1. Items 1-7 of the [Coding Standard](#) will be enforced for this project.
2. You are not allowed to use advanced data structures such as lists or dictionaries.
3. My solution has multiple loops, multiple `if` statements, string concatenation, `len()`, slicing, and `in`. I also used `and` and `not` in Booleans, but your logic may be different.
4. How do you “collect vowels”? Start with an empty string and concatenate each vowel onto it:
`vowels += ch`
Similar for consonants. Collect in the same order that they appear in the word (necessary to match the Mimir tests).
5. How do you check to see if a character is a vowel? Define a constant string (at the top of your program using all caps) and then check with an `if` statement using in:
`VOWELS = "aeiou"`
`if ch in VOWELS: # True if ch is a vowel`
`# do something`
6. There are many ways to solve this problem. I found `enumerate()` helpful to find the index of the last vowel.
7. Where did I use `and` in a Boolean? One place was in my main `while` loop where I kept looping while I did not have enough vowels and I did not have enough consonants.

8. To help when debugging I printed out the vowel string each time I added to it and did the same with consonants. Once everything was working I simply commented them out (or you could delete them).
9. Here is the formatting for my table header (and my string of equal signs). Not only is the formatting useful for the header, but it provides clues on how to format the data. You want to get the formatting exactly correct so you will pass the Mimir tests.

```
print("\n"+"="*12)
print("{:8s}{:7s} | {:12s}{:7s}".format("vowels","length","consonants","length"))
```

Getting Started

- *Solve the problem using pencil and paper first.* You cannot write a program until you have figured out how to solve the problem. This first step can be done collaboratively with another student. However, once the discussion turns to Python specifics and the subsequent writing of Python, you must work on your own.
- Use Anaconda Spyder to create a new program. Use the required file name (`proj03.py`).
- Write a simple version of the program, e.g. create a while loop that stops when all five vowels have been collected; within the loop input a character (string of length 1) and print the character. Run the program and track down any errors.
- Use the **Mimir** system to turn in the first version of your program. All tests will fail initially because we aren't far enough to pass the first test.
- Next input a word and then loop through the word to collect vowels in the word. Now you have a loop within a loop.
- Next handle consonants. You need to find the index of the final vowel. I modified my vowel-finding loop to use `enumerate()` to get each character and its index. For example, `for i,ch in enumerate(in_str):`
- Cycle through the steps to incrementally develop your program:
 - Edit your program to add new capabilities.
 - Run the program and fix any errors.
- Use the **Mimir** system to submit your final version.

Sample Interaction

Test 1 (terminates when all vowels are found)

```
Input a word: hello
Input a word: luck
Input a word: liar
=====
```

vowels	length		consonants	length
euia	5		ckr	3

Test 2 (terminates when number of consonants is 5 or more)

Input a word: string

Input a word: hits

Input a word: clock

=====

vowels	length		consonants	length
io	2		ngtsck	6

Test 3 (terminates on vowels and there are no trailing consonants in one word)

Input a word: sequoia

=====

vowels	length		consonants	length
euoia	5			0

Test 4 (terminates on trailing consonants in one word)

Input a word: bimonthly

=====

vowels	length		consonants	length
io	2		nthly	5

Test 5 (multiple occurrences of the same consonant)

Input a word: a

Input a word: be

Input a word: spaciousness

=====

vowels	length		consonants	length
aeiou	5		s	1

Scoring Rubric

Computer Project #3 Scoring Summary

General Requirements

__0__ (3 pts) Coding standard

Program Implementation

__0__ (4 pts) Test case 1

__0__ (4 pts) Test case 2

__0__ (3 pts) Test case 3

__0__ (3 pts) Test case 4

__0__ (3 pts) Test case 5