

**HOMEWORK 4****(submit all .py files as a single .zip file to the Wolfware course site)**

- 1) Write the following functions. Each of these functions should have a single parameter -- accepting a string as an argument. The function should **only** do what is specified:
  1. Write a function that counts and returns the number of vowels in the string (A, E, I, O, U, a, e, i, o, u)
  2. Write a function that counts and returns the number of consonants in the string.
  3. Write a function that converts the string to all lowercase.
  4. Write a function that converts the string to all uppercase.
  
- 2) Write a main program that performs the following steps:
  - A. Prompt the user to enter a string, and let them type it in. This could be an entire sentence, with the newline indicating the end of the string.
  
  - B. Display the following menu:
    - A Count the number of vowels in the string
    - B Count the number of consonants in the string
    - C Convert the string to uppercase
    - D Convert the string to lowercase
    - E Enter another string
  
    - M Display this menu
    - X Exit the program
  
  - C. Enter a loop, allowing the user to type in a menu choice from above each time. Loop should continue until the user enters the command to exit. Upper and lowercase letters should be allowed for the menu choices.
    - a. When the A or B commands are entered (counting vowels or consonants), call the corresponding function, then print the result
    - b. When C or D menu options are chosen, just call the appropriate function to convert the string and print the result to the console.
    - c. When E is chosen, allow a new string to be typed -- this will replace the previous one. And the menu should be re-displayed back to the console.
    - d. The menu should only be displayed once at the start, and then again whenever the E or M option is selected
  
- 3) Write a python to accept the birth date of the user (in mm/dd/yyyy format) and calculate the number of days between current system date and the birth date. Ensure that the user has selected a date prior to system date. Check if the user has entered it in the correct format including the '/' within the string. Print errors if any to the screen. If errors are seen, go back to asking the user to reenter the date.
  
- 4) The future value of money, relates how much a current investment will be worth in the future, assuming a constant interest rate.

$$FV = PV * (1 + Int)^n$$

Where FV = future value; PV = Present value; Int = interest rate per compounding period expressed as a fraction, ex 5% as 0.05; n = compounding periods

Write a Python function to calculate FV with the three inputs (PV, Int, n). Dr Starly has three options to decide on where to invest his \$10,000, as given by choice options in the table. Generate an output that displays the total value in his account every year until the maturity year for each of the 3 institutions.

Financial Institution	Int. Rate per month	Years
NC Capital Bank	2.3%	3
Mutual funds	2.6%	5
Allie Bank	2.5%	4

- 5) Consider the behavior of a freely falling object under the influence of gravity. The position of the object is given by

$$d = 0.5 * g * t^2$$

d = distance object travels; g= acceleration due to gravity; t = elapsed time.

The script can set the value of 'g' assuming Earth's gravity. Ask the user to enter the start time, end time and the desired time increments. Print the value of d, during each time step. Use consistent system of units.

- 6) Given a list of students with the following scores received as part of the Python class. Write a function that will return the alphabetical grade for the following grading scale: A: >=90, B:80-89, C: 70-79, D: 60-69, F: <59. Implement a numeric look up table. Your code script **MUST NOT** have a series of IF or IF ELSE conditions that assigns the letter grade based a range check. For example, these statements are **not allowed**:

```
If (score>80) and (score<89):
    Grade='A'
```

```
If (score<50):
    Grade='F'
```

The following input data can be used formatted as a list of dictionaries with Name and Score as the keys.

Name	Score
Binil	90
Starly	40
Ethan	88
Evan	87
Rachel	95
Michele	79
Jayden	82
Ben	48

The result must be in the following format as a list of tuples. Do not use special libraries other than the standard python. Use of math library is allowed.

[(Name, Score, Letter Grade) . . . . .]