Homework Three

• (Ungraded) Ch 7 T/F, Ch 9 Multiple Choice 1-8, Ch 5 Multiple Choice 10 -17, Ch 12 T/F. 1-2 quiz questions will be from these questions.

• (Ungraded) Complete the Datacamp functions chapter, and the lists chapter.

• (Graded) Gaddis Exercises 5.5, 5.19, 7.8, 7.12, 9.10, 12.7. Submit one document upload LastnameFirstname-hw3.py (e.g. VasilakyKathryn-hw3.py)

Important Information About Exercises

The following exercise descriptions are followed by a short description of requirements. Your code will be graded with a script which will check for your function names, and output to be *exactly* as described in this section.

Write your code in your own file, and define functions with the names provided in this document, if you fail to do so your code *will fail* the script being used to grade it. Note that for all prompts, it is assumed that a space should come after the prompt text to allow for user input. Also note that all output should end with a newline (the next shell prompt is located beneath the text) unless instructed otherwise. For example if the "hello world" program was one of the exercises it would be listed as follows:

Requirements

- Function Name:

  ```
  hello_world
  ```

- Output with sample input "hello":

  ```
  Say hello to the computer: hello
  hello, world
  ```

Your code, written in your own file, would have to look like this:

```
def hello_world():
    input("Say hello to the computer: ")
    print("hello, world")
```

5.5. Property Tax

A county collects property taxes on the assessment value of property, which is 60 percent of the property's actual value. For example, if an acre of land is valued at $10,000, its assessment value is $6,000. The property tax is then 72¢ for each $100 of the assessment value. The tax for the acre assessed at $6,000 will be $43.20. Write a program that asks for the actual value of a piece of property and displays the assessment value and property tax.

Requirements

- Function Name:

```
property_tax
```

- Output with sample input "10000":

```
Enter the actual value: 10000
Assessed value: $6,000.00
Property tax: $43.20
```

5.9 Future Value

Suppose you have a certain amount of money in a savings account that earns compound monthly interest, and you want to calculate the amount that you will have after a specific number of months. The formula is as follows:

$$F = P \times (1 + i)^t$$

The terms in the formula are:

F is the future value of the account after the specified time period.

P is the present value of the account.

i is the monthly interest rate.

t is the number of months.

Requirements

- Function Name:

```
future_value
```

- Output with sample input "100", "5", and "12":

```
Enter the present value of the account in dollars: 100
Enter the monthly interest rate as a percentage: 5
Enter the number of months: 12
The information for your account is:
Present value: $100.00
Interest Rate: %5.00
After 12 months, the value of your account will be $179.59
```

7.8 Generation Z Search

You will find the following files in the Hw3 folder:

GirlNames.txt This file contains a list of the 200 most popular names given to girls born in the United States from the year 2000 through 2009.

BoyNames.txt This file contains a list of the 200 most popular names given to boys born in the United States from the year 2000 through 2009.

See the lecture notes for starter code on how to read in txt files.

Write a program that reads the contents of the two files into two separate lists. The user should be able to enter a boy's name, a girl's name, or both, and the application will display messages indicating whether the names were among the most popular.

Requirements

- Function Name:

```
genz_search
```

- Output with sample input "Kayne", and "Kim":

```
Enter a boy's name, or N if you do not wish to enter a boy's name: Kayne
Enter a girl's name, or N if you do not wish to enter a girl's name: Kim
Kayne is not one of the most popular boy's names.
Kim is not one of the most popular girl's names.
```

- Output with sample input "N", and "N":

```
Enter a boy's name, or N if you do not wish to enter a boy's name: N
Enter a girl's name, or N if you do not wish to enter a girl's name: N
You chose not to enter a boy's name.
You chose not to enter a girl's name.
```

7.12 Prime Number Generation

A positive integer greater than 1 is said to be prime if it has no divisors other than 1 and itself. A positive integer greater than 1 is composite if it is not prime. Write a program that asks the user to enter an integer greater than 1, then displays all of the prime numbers that are less than or equal to the number entered. The program should work as follows:

Once the user has entered a number, the program should populate a list with all of the integers from 2 up through the value entered.

The program should then use a loop to step through the list. The loop should pass each element to a function that displays the element whether it is a prime number.

Requirements

- Function Name:

```
prime_gen
```

- Output with sample input "10":

```
Enter an integer greater than 1: 10
2 is prime.
3 is prime.
4 is composite.
5 is prime.
6 is composite.
7 is prime.
8 is composite.
9 is composite.
10 is composite.
```

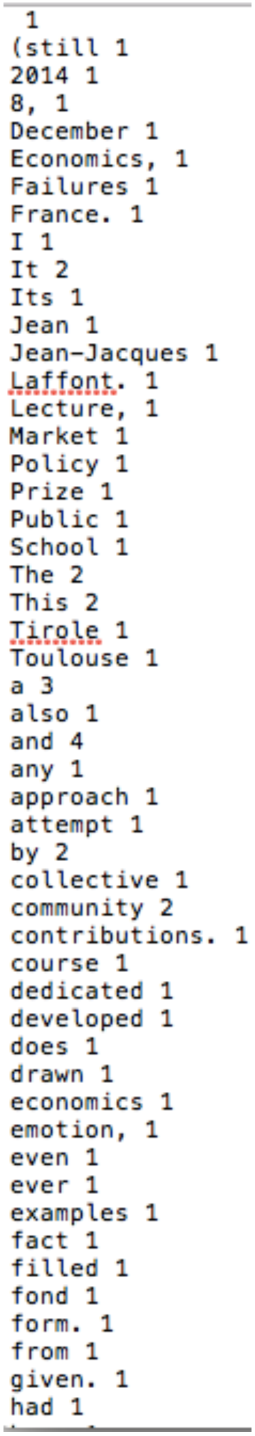
9.10 Jean Tirole Nobel Index

Write a program that reads the contents of a Jean Tirole's Nobel prize 2014 winning speech in economics for his work on market failures and public policy (tirole_nobel.txt). The program should create a dictionary in which the key-value pairs are described as follows:

Key. The keys are the individual words found in the file.

Values. Each value is the number of times that word is found in the text.

Once the dictionary is built, the program should create another text file, index_j.tx, listing the contents of the dictionary. The word index file should contain an alphabetical listing of the words that are stored as keys in the dictionary, along with the count of the words appear in the original file.



```
 1
(still 1
2014 1
8, 1
December 1
Economics, 1
Failures 1
France. 1
I 1
It 2
Its 1
Jean 1
Jean-Jacques 1
Laffont. 1
Lecture, 1
Market 1
Policy 1
Prize 1
Public 1
School 1
The 2
This 2
Tirole 1
Toulouse 1
a 3
also 1
and 4
any 1
approach 1
attempt 1
by 2
collective 1
community 2
contributions. 1
course 1
dedicated 1
developed 1
does 1
drawn 1
economics 1
emotion, 1
even 1
ever 1
examples 1
fact 1
filled 1
fond 1
form. 1
from 1
given. 1
had 1
```

Requirements

- Function Name: Use the names specified in the provided template file.
- Output: No shell output, look at the sample images above for what file output should look like

12.7 Recursive Power Method

Design a function that uses recursion to raise a number to a power. The function should accept two arguments: the number to be raised, and the exponent. Assume the exponent is a nonnegative integer.

Requirements

- Function Name:

  ```
  pow_recursive
  ```

- Output with sample input "10":

  ```
  Enter a positive whole number between 1 and 100: 5
  10.0 raised to the power of 5 is 100,000.00
  ```