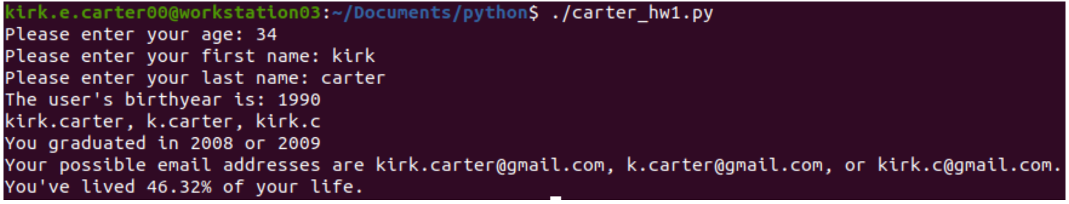
Basic [NOT] Homework :)

Homework 1

Concepts covered:
- User input
- String concatenation
- String formatting
- Variable assignment
- Comments
- Mathematical Operators
- Floats

Create a program that accepts the age, first name, and last name of the user as user input.
The program should then print:
1. A sentence that states the user's birthyear.
2. 3 possible username formats based on their first name and last name.
3. 2 possible years the user might have graduated high school.
4. 3 possible "@gmail.com" email addresses based on the previous identified username formats.
5. Percentage of life completed based on life expectancy of 73.4.

General requirements for the program:
1. Shebang so it can be run by executing the file
2. Use both string concantenation and string formatting at least once.
3. Use appropriate variables to assign your values.
4. Use appropriate comments to explain how your program is functioning.
5. Name your file "lname_hw1.py"



```
kirk.e.carter00@workstation03:~/Documents/python$ ./carter_hw1.py
Please enter your age: 34
Please enter your first name: kirk
Please enter your last name: carter
The user's birthyear is: 1990
kirk.carter, k.carter, kirk.c
You graduated in 2008 or 2009
Your possible email addresses are kirk.carter@gmail.com, k.carter@gmail.com, or kirk.c@gmail.com.
You've lived 46.32% of your life.
```

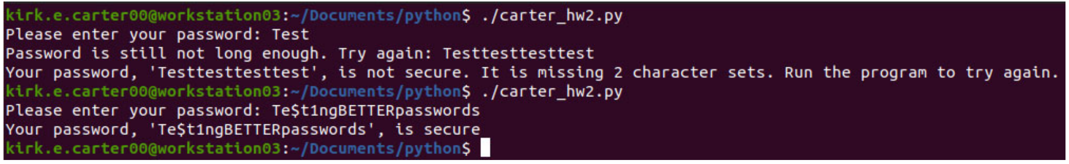
Homework 2

Concepts covered:
- Creating a function
- User input
- Conditional check
- While

- If / else
- String methods

Create a **function** to take a password as **user input**. Using a **while loop**, **if** the password is less than 14 characters, prompt the user for a password until the user input is 14 or more characters. Once the user types in a password of 14 or more characters, check the password for an uppercase, lowercase, digit, and special character. If the user has all elements, print out that the password is secure, **else** print out password is not secure.

Please see the below image as a sample of what your program could look like. You do not have to specify how many or which character sets are missing. I just wanted to track it during my testing.



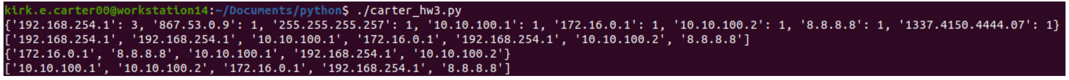
```
kirk.e.carter00@workstation03:~/Documents/python$ ./carter_hw2.py
Please enter your password: Test
Password is still not long enough. Try again: Testtesttesttest
Your password, 'Testtesttesttest', is not secure. It is missing 2 character sets. Run the program to try again.
kirk.e.carter00@workstation03:~/Documents/python$ ./carter_hw2.py
Please enter your password: Te$t1ngBETTERpasswords
Your password, 'Te$t1ngBETTERpasswords', is secure
kirk.e.carter00@workstation03:~/Documents/python$ 
```

Homework 3

Concepts covered:
- dictionaries
- set
- list
- iterable methods
- conditionals
  - Simple checking OR
  - Regular expressions
- For loops

addresses = (
'192.168.254.1',
'867.53.0.9',
'192.168.254.1',
'255.255.255.257',
'10.10.100.1',
'172.16.0.1',
'192.168.254.1',
'10.10.100.2',
'8.8.8.8',
'1337.4150.4444.07'
)

Write a program that uses the above tuple of IP addresses to:
Print a dictionary with each IP address as the key and the count of occurences of each IP address in the tuple as its value.
Evalute each IP address for validity.
Print a list of all valid IP address occurences in order of appearance.
Print a set of all valid IP addresses.
Print a sorted set of all unique valid IP addresses.



```
kirk.e.carter00@workstation14:~/Documents/python$ ./carter_hw3.py
{'192.168.254.1': 3, '867.53.0.9': 1, '255.255.255.257': 1, '10.10.100.1': 1, '172.16.0.1': 1, '10.10.100.2': 1, '8.8.8.8': 1, '1337.4150.4444.07': 1}
['192.168.254.1', '192.168.254.1', '10.10.100.1', '172.16.0.1', '192.168.254.1', '10.10.100.2', '8.8.8.8']
{'172.16.0.1', '8.8.8.8', '10.10.100.1', '192.168.254.1', '10.10.100.2'}
['10.10.100.1', '10.10.100.2', '172.16.0.1', '192.168.254.1', '8.8.8.8']
```

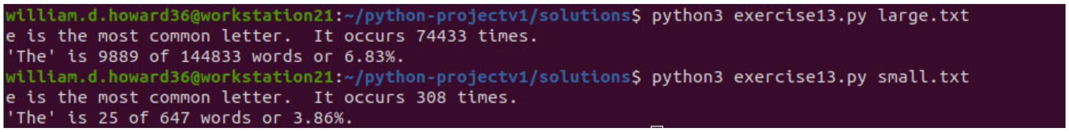
Homework 4

Attached Files: large.txt (826.472 KB)
small.txt (3.511 KB)

Concepts covered:
- file input/output
- string formatting
- string methods
- mathematical operations

Create program that:

- Reads a filename from the command line.
- Determines the most common letter in the file that is not the white space characters, and prints what it is and how many times it occurs. Use the format " is the most common letter. It occurs _ times." Replace _ with the appropriate letter (uppercase) and number. **Case is important.**
- Determines what percentage of the number of words in the file is the word "the"; print the percentage to two points of decimal precision. **Ignore capitalization**: "The" and "the" are the same word.
- Writes the first ten words of the file (as determined by whitespace) to a new file named "ten_words.txt". Assume the file will be written to the same directory where your program is located.

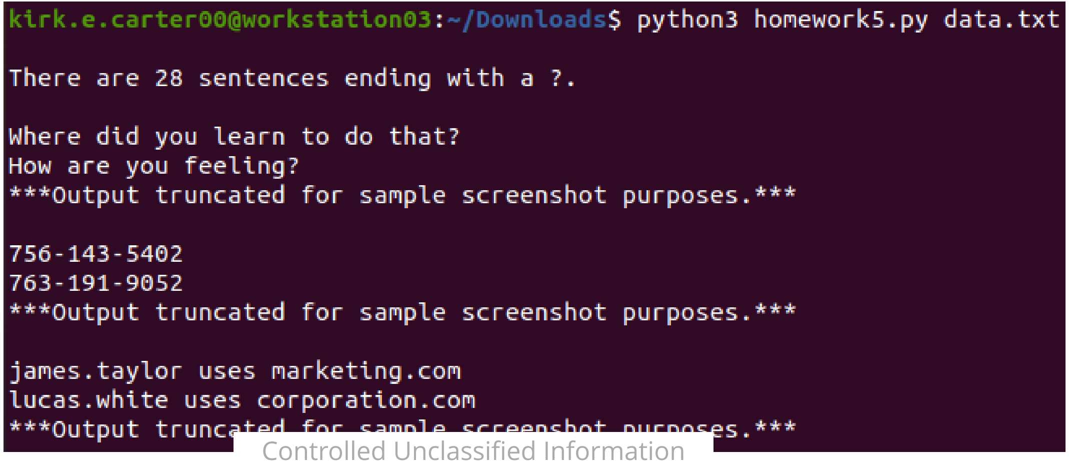


```
william.d.howard36@workstation21:~/python-projectv1/solutions$ python3 exercise13.py large.txt
e is the most common letter.  It occurs 74433 times.
'The' is 9889 of 144833 words or 6.83%.
william.d.howard36@workstation21:~/python-projectv1/solutions$ python3 exercise13.py small.txt
e is the most common letter.  It occurs 308 times.
'The' is 25 of 647 words or 3.86%.
```

Homework 5

Concepts covered:
- regular expressions
- string splitting OR
- capture groups

Write a python program that takes a file name as a command line argument that will:
1. Counts the number of lines ending in a '?'. You will print that value, followed by all of the sentences.
2. Find all phone numbers and print them to the screen. Format for phone numbers should all be 706-123-4567.
3. Find all ".com" email addresses in the file. Print their username and the email domain. Format should be: kirk.carter uses gmail.com



```
kirk.e.carter00@workstation03:~/Downloads$ python3 homework5.py data.txt

There are 28 sentences ending with a ?.

Where did you learn to do that?
How are you feeling?
***Output truncated for sample screenshot purposes.***

756-143-5402
763-191-9052
***Output truncated for sample screenshot purposes.***

james.taylor uses marketing.com
lucas.white uses corporation.com
***Output truncated for sample screenshot purposes.***
```