Assignment 8

Use the following naming scheme for your program files: `a`*assignment#*`p`*problem#yourname*`.py` . So your program for problem 1 on this assignment will be named `a8p1bob.py` and your for problem 2 will be named `a8p2bob.py` (adjust to be your name). Please submit all of your .py files to the Moodle dropbox.

Problems

1. ### Text file practice

   Write a module containing three functions. When run, the module should prompt the user for a filename and a number of lines (let's refer to it as `n`) and then call each function in turn with these parameters.

   a. The function `head` should return a list containing the first `n` lines of the file.

   b. The function `tail` should return a list containing the last `n` lines of the file.

   c. The function `longest` should return a list containing the longest `n` lines of the file.

   The output of running the module might look like this:

   ```
   Name of file to test with: pooh.txt
   Number of lines to display: 3

   head says the first 3 lines are:
   The more it snows
     (Tiddely pom),
   The more it goes

   tail says the last 3 lines are:
   ```

```
 How cold my toes
  (Tiddely pom),
 Are growing.

longest says the longest 3 lines are:
The more it snows
The more it goes
And nobody knows
```

Since the file could be very long you will **not** want to read the whole thing into memory as a list of lines...

2. Testing `biggest()`

The last few assignments have provided you with examples of testing harnesses to ensure programs are working correctly. Now it is time for you to design some tests of your own. The file a8_2.py contains six versions of a function named `biggest` that all claim to return the largest of three values they are passed. Add test cases to the program to determine which ones (if any) work correctly. Based on your test cases identify the situations (if any) in which each version fails, and the reason for the failure, i.e. what the problem is with the code. Enter your answers into the docstrings for each function.

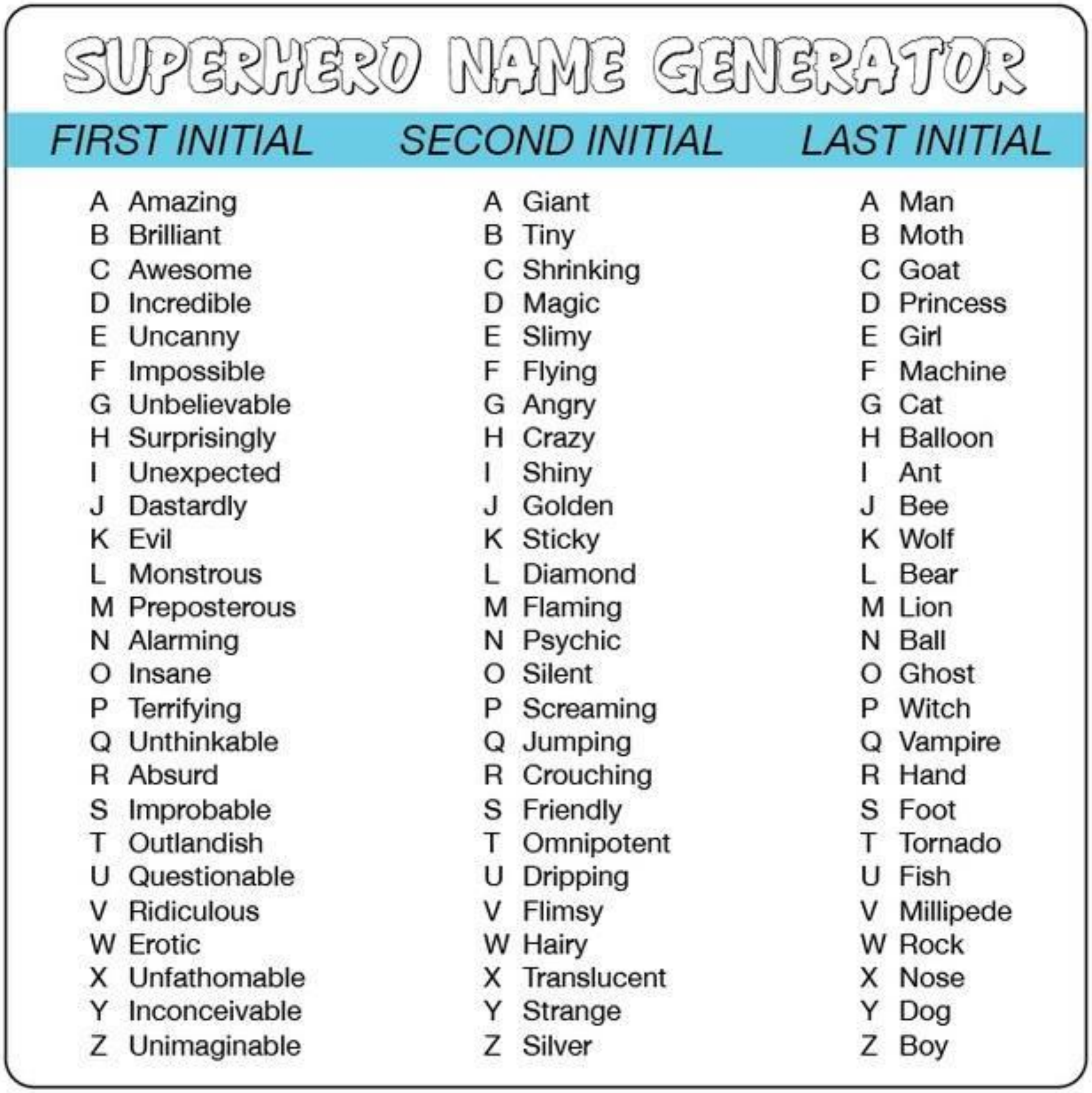
3. CRUD Control

Complete the program crud_controller.py, i.e. insert code for the remaining functions, and make sure adding new quotes can't delete existing ones. Rename it a8p3*name*.py.

4. OPTIONAL BONUS: Super hero name generator

Write a function called `superhero_name_generator` that is passed a string containing someone's full name, e.g. `'Timothy Nicholas Topper'`, and returns their superhero name according to the chart below, e.g. my superhero name is Outlandish Psychic Tornado.

Put your function in a module with an `if __name__ == '__main__':` section that tests it.



SUPERHERO NAME GENERATOR

FIRST INITIAL		SECOND INITIAL		LAST INITIAL	
A	Amazing	A	Giant	A	Man
B	Brilliant	B	Tiny	B	Moth
C	Awesome	C	Shrinking	C	Goat
D	Incredible	D	Magic	D	Princess
E	Uncanny	E	Slimy	E	Girl
F	Impossible	F	Flying	F	Machine
G	Unbelievable	G	Angry	G	Cat
H	Surprisingly	H	Crazy	H	Balloon
I	Unexpected	I	Shiny	I	Ant
J	Dastardly	J	Golden	J	Bee
K	Evil	K	Sticky	K	Wolf
L	Monstrous	L	Diamond	L	Bear
M	Preposterous	M	Flaming	M	Lion
N	Alarming	N	Psychic	N	Ball
O	Insane	O	Silent	O	Ghost
P	Terrifying	P	Screaming	P	Witch
Q	Unthinkable	Q	Jumping	Q	Vampire
R	Absurd	R	Crouching	R	Hand
S	Improbable	S	Friendly	S	Foot
T	Outlandish	T	Omnipotent	T	Tornado
U	Questionable	U	Dripping	U	Fish
V	Ridiculous	V	Flimsy	V	Millipede
W	Erotic	W	Hairy	W	Rock
X	Unfathomable	X	Translucent	X	Nose
Y	Inconceivable	Y	Strange	Y	Dog
Z	Unimaginable	Z	Silver	Z	Boy