Problem A. File of Names [50 points]

Problem

The file `Names.txt` contains a list of first names in alphabetical order. Write a program that requests a name from the user and inserts the name into the file in its proper location. If the name is already in the file, it should not be inserted. You should store the data from the file as the set data type, and use set operations in your program.

Functions

- `readSetFromFile()`: read set from `Names.txt`. If the file does not exist, exit.
- `inputName()`: input the name from the terminal.
- `insertSet(mySet, name)`: insert the name into set.
- `writeToFile(modifiedSet)`: write set to `Names.txt`.

Restrictions

- You should use `set` operations in your program.
- Use os.path.isfile method and exit function in readSetFromFile().
- Do not modify main function and other function names.

Skeleton Code

```
import os.path

def main():
        mySet = readSetFromFile()
        name = inputName()
        modifiedSet = insertSet(mySet, name)
        writeToFile(modifiedSet)

def readSetFromFile():                  # implement functions
def inputName():
def insertSet(mySet, name):
def writeToFile(modifiedSet):

main()
```

Example I/O

- You should write `Names.txt` in your own to execute the examples.

Terminal	Names.txt (Before)	Names.txt (After)

`Enter a first name to be included: Grape`
`Grape is added in Names.txt`
	`Apple`
	`Orange`
		`Apple`
		`Grape`
		`Orange`

`Enter a first name to be included: Grape`
`Grape is already in Names.txt`
	`Apple`
	`Grape`
	`Orange`
		`Apple`
		`Grape`
		`Orange`

`Names.txt does not exist.` `Terminate program.`		

Submit format

- HW03_A_(NAME).py

Problem B. Unit Conversions [50 points]

Problem

The following table contains some lengths in terms of feet. Write a program that requests the unit to convert from, the unit to convert to, and the quantity to be converted; and then displays the converted quantity. Use the file `Units.txt` to create a dictionary that provides the number of feet for a given unit of length. The first two lines of the file are `inch,.083333;yard,3.`

Equivalent lengths.

1 inch = .083333 foot	1 rod = 16.5 feet
1 yard = 3 feet	1 furlong = 660 feet
1 meter = 3.28155 feet	1 kilometer = 3281.5 feet
1 fathom = 6 feet	1 mile = 5280 feet

Functions

- `populateDictionary()`: create dictionary from `Units.txt` to convert units.
- `getInput()`: input units and length from the terminal.

Restrictions

- You should use `dictionary` operations in your program.
- Do not modify main function and other functions names.

Skeleton Code

```
def main():
    feet = populateDictionary()
    orig, dest, length = getInput()
    ans = length * feet[orig] / feet[dest]
    print("Length in {0}: {1:,.4f}".format(dest, ans))

# implement functions
def populateDictionary():
def getInput():

main()
```

Example I/O

```
Unit to convert from: yard
Unit to convert to: mile
Enter length in yard: 555
Length in mile: 0.3153
```

Submit format

- HW03_B_(NAME).py