COP 2930 Sample Final Exam Part B: Coding Problems

Directions: Write a single python program to solve each problem posed. No main function necessary. For questions that ask to write a function, write the function and test it with the main function provided. The name for your file will be provided within the parentheses at the beginning of the question.

1) (**milk.py**) Write in a python program that reads in how many gallons of milk are in the refrigerator (as an integer), and prints out the equivalent number of ounces. (Note: One gallon equals 128 ounces.)

2) (**money.py**) Scholarships are award to all students who have a 3.0 GPA or higher and have done 100 or more community service hours. (Both requirements have to be met to earn the scholarship.) Write a program that asks the user their GPA (reads it in as floating point number) and the number of hours of community service they've done (as an integer) and outputs whether or not the user receives the scholarship.

3) (**textx.py**)Write a program that reads in a single positive odd integer n and prints out a text 'X' of that many rows using the star character. You may assume that a positive odd integer is entered. Here is the design that should be drawn for n = 5:

```
*     *
 *   *
  * *
   *
  * *
 *   *
*     *
```

Note: in this design there are 5, 3 and 1 spaces respectively between the two stars on the first three lines, and 1, 3 and 5, spaces respectively between the two stars on the last three lines.

4) (**book.py**) Write a function, `sheetsNeeded`, that takes in the number of pages in a book and returns the number of 11" x 17" sheets of paper necessary to print the book. Note, a single 11" x 17" sheet of paper can be used to print up to 4 pages of a book. For example, if the input to the function is 98, then the function should return 25, since 24 sheets would only allow 96 pages but 25 sheets allows up to 100 pages. To get full credit, you can NOT use a loop in your function. Please include the following lines of code for testing in your submission. Note that I may change the tests for grading purposes but wanted to provide some to you due to the time constraint.

```
def sheetsNeeded(pages):
      # Put your code here

def main():
    print(sheetsNeeded(100))
    print(sheetsNeeded(98))
    print(sheetsNeeded(123456789))
    print(sheetsNeeded(48687))

main()
```

5) (**names.py**) The first line of the file, names.txt, contains a single positive integer, *n*, representing the number of names that follow, in the file. The following *n* lines each contain one name. Each name will be unique and consist only of uppercase letters. Write a program that reads in all of this information from a file and stores this information in a dictionary, where each name is mapped to the number of letters in it. Please call the dictionary, *nameDictionary*. You may test your program with the file shown below and the code shown after that:

Sample File (names.txt)

```
10
DAVID
AARON
ANNABELLE
QINGQING
YOLANDA
JORDANA
BRUCE
YI
BEN
CONSTANTINE
```

Testing Code for this file:

```
names = ["BEN","ANNA","DAVID","ELAINA","JORDANA","YOLANDAS"]
for name in names:
    if name in nameDictionary:
        print("This name has",nameDictionary.get(name),"letters")
    else
        print("Not in dictionary.")
```