Test 1 Redux/Booster: Compsci 06

October 20, 2010

Name: ____

NetID/Login: _____________

Community Standard Acknowledgment (signature) ______________________________________

	value	grade
Problem 1	12 pts.	
Problem 2	12 pts.	

In writing code you do not need to worry about specifying the proper `import statements`. Don't worry about getting function or method names exactly right. Assume that all libraries and packages we've discussed are imported in any code you write.

PROBLEM 1 : (***Phunkadelic (12 points)***)

Part A

A number is *square free* if it is not divisible by any perfect square greater than one. For example, 10 is square free since it is not divisible by four nor by nine, the two perfect squares less than 10. The number 100 is **not square free** since it is divisible by 25, and 25 is a perfect square and by 4 which is also a perfect square. Write the function `isSquareFree` to return True if its int parameter is square free, and false otherwise. For example:

call	**return value**
`isSquareFree(8)`	`False`
`isSquareFree(45)`	`False`
`isSquareFree(38)`	`True`
`isSquareFree(55)`	`True`

Hint: if you loop over 1,2,3,4, . . . you can test divisors 1,4,9,16, . . . by squaring each of the 1,2,3,4 being looped over.

```
def isSquareFree(num):
    """
    return True if int parameter num is square free and
    returns False otherwise
    """
```

Part B

Some words contain other words. For example, each of "sublime", "compliment", "limerick" and "millimeter" contains the word "lime". Write the function wordCount that returns the number of strings in its list parameter words that contain the string sub.

call	return value
wordCount(["sublime", "millimeter", "lemon"],"lime")	2
wordCount(["subtract", "assume", "consumer","presume","lime"],"sum")	3
wordCount(["apple", "banana", "lemon"],"meat")	0

```
def wordCount(words,sub):
    """
    return the number of strings in string list words
    that contain string sub
    """
```

PROBLEM 2 : (***Genus, Order, Class, . . .***)

Data is stored in a file in the format shown below. Each line contains data for one animal giving the animal's name (string), gestation period in days (int), and estimated longevity in years (int). The information on a line is delimited by commas as shown, for example the file below shows information for eight animals in the format used in this problem.

```
bear,180,15
cat,52,10
dog,53,10
hamster,15,2
elephant,510,30
hippopotamus,220,30
human,253,65
lion,106,10
```

Write the function `getAgeList` that returns a list of those animals whose estimated longevity is between the values given by its two int parameters: `low` and `high`. The name of the file holding the data to be read and processed is given by parameter `filename`.

For example, if `"data.txt"` is the name of the sample data file above, then the call `getAgeList("data.txt",15,30)` should return the list `["bear","elephant","hippopotamus"]`, the call `getAgeList("data.txt",1,8)` should return the list `["hamster"]` and the call `getAgeList("data.txt",70,100)` should return the empty list `[]`

```
def getAgeList(filename, low, high):
    file = open(filename);









    file.close()
```