CompSci 101: Test 2

Peter Lorensen

April 13, 2013

Name: _____

NetID/Login: _____

Community Standard Acknowledgment (signature): _____

	Value	Grade
Problem 1.	26 pts	
Problem 2	32 pts.	
Problem 3	32 pts.	
TOTAL:	90 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 : (RENTAL FLEET 26 POINTS)

You will be asked to write code that refers to the list below. The python code you write should work with any values stored in the list.

```
rentalFleet = [ "MazdaCX-5",  "FordTaurus", "Mazda5", "Ram2500", "BuickEncore",
                "MazdaCX-5", "Mazda5", "Ram2500", "Ram2500", "MazdaCX-5",
                "FordTaurus", "LexusLS", "MazdaCX-5", "Ram2500"  ]
```

Part A (4 points)

Fill up the new list named ford with strings that begins with "Ford".

Part B (4points)

Write code that stores in a variable difCar the number of different cars that the list rentalFleet has.

Part C (8 points)

Write code that stores in a variable most the most frequent occurring car in rentalFleet. In case of a tie, sort alphabetically.

Part D (10 points)

Sort the list rentalFleet putting the most frequently occurring car first (do not worry about breaking ties). However, you must first remove all car names with a digit in it (i.e. a number 0-9, so that for example "Ram2500" must be taken out). The list must only contain unique cars.

PROBLEM 2 : (MOVIE RENTALS 32 POINTS)

You will be asked to write code that refers to the list below. The python code you write should work with any values stored in the list. You may assume that functions written in one part of this problem solution will work perfectly when used in other parts of the solution.

Given the following data from a movie rental company:

```
movieData = [ [“Alien”, 2.50, (“Arguson”, False)],
              [“Codebreaker”, 2.00, (“Johnson”, True)],
              [“Ice Age 4”, 1.45, (“Yee”, True)],
              [“Ice Age 4”, 1.45, (“Miller”, False)],
              [“Transformers”, 0.99, (“Miller”, False)],
              [“Alien 2”, 3.50, (“Arguson”, False)],
              [“Alien 3”, 4.50, (“Arguson”, False)],
              [“Codebreaker”, 2.00, (“Yee”, True)],   ]
```

It contains: [[“movie name”, price, (username, premium customer)]]

It is a list of lists where each entry contains the movie name, the price of the movie and then a tuple with the renterøs name and a boolean that is set to True if the renter is a premium customer. The type of each item is:

[[string, float, (string, boolean)]]

Part A (6 points)

Write code that return a list of tuples like this (the order is irrelevant):

```
[(“Arguson”, False), (“Johnson”, True), (“Yee”, True), (“Miller”, False), (“Miller”, False),
 (“Arguson”, False), (“Arguson”, False), (“Yee”, True)]
```

The method should return the user information from the movieData list. The type is [(username, premium customer)]:

```
def getUserInfo( movieData ):
```

Part B (6 points)

Assume that your function in Part A works perfectly, write code to find the total number of premium customers:

```
def getNumPremiums( movieData  ):
```

Part C (10 points)

Write code to return a dictionary like this:

```
{ “Alien”: [“Arguson”], “Codebreaker”: [“Johnson”, “Yee”], “Ice Age 4”: [“Yee”,
  “Miller”], “Transformers”: [“Miller”], “Alien 2”: [“Arguson”], “Alien 3”: [“Arguson”] }
```

The method should return a dictionary of all the movies with the usernames of who has rented it (note if a user has rented the same movie twice his/her name should appear twice):

```
def getMovieInfo( movieData ):
```

Part D (10 points)

Write code to find the total amount of money that each user has spent. The function must return a dictionary like this:

```
{ “Arguson”: 10.5, “Johnson”: 2.0, “Yee”: 3.45, “Miller”: 2.44 }
```

```
def getUserTotal( movieData ):
```

PROBLEM 3: (BAR ALIANCE 32 POINTS)

You will be asked to write code that refers to the list below. The python code you write should work with any values stored in the list. You may assume that functions written in one part of this problem solution will work perfectly when used in other parts of the solution.

In a city the bar guests often change bars during a night out in town, going to several different bars to meet friends.

A group of bars forms an alliance and hands out Alliance Cards that has a number to identify the customer. The Alliance Card gives customers cheaper drinks the more alliance bars they visit.

The following data is a list of all the bars in the alliance with all the customer numbers that have visited one night. Each sub list represents a bar and bar nr. 0 has index 0, bar nr. 1has index 1 and so on. Each sub list contains Alliance Card numbers – one number for each person visiting the bar that evening.

```
allBars = [  [“10245”, “83762”, “33736”], [“10245”, “55675”],
             [“71221”, “10245”, “83762’, “41230”], [“55675”, “83762”, “10245”]  ]
```

Part A (8 points)

Write code for this method that returns the total number of unique bar customers:

```
def totalBarGuests( allBars ):
```

Part B (12 points)

Write code for the method that returns a list of customer numbers who visited more than one bar. For the example above the, method should return this list:

```
["10245", "83762", "55675"]
```

```
def getBarHoppers( allBars ):
```

Part C (12 points)

You may use any methods written in above and assume that it works correctly. Write code for this method that returns a list of tuples that contains the following:

```
[  (index of bar in allBars, number of guests visiting more than just this bar,
    number of guests visiting ONLY this bar)  ]
```

For the example allBars above the method should return this list:

```
[ (0, 2, 1), (1, 2,0), (2, 2, 2), (3, 3, 0) ]
```

```
def getBarGuests( allBars ):
```

(Extra page for writing code)