**11/5/15**

"""

Donato, Brandon

bdonato1@binghamton.edu

CS 110 - B57

Jia Yang

Assignment9Ex1

"""

"""

Analysis:

Write a program that prints out a table showing the number of scores and score

average for each person, generated two different ways, given a literal list of

tuples consisting of (name, list of grades) pairs.

Output to monitor:

key - Name, which is a key that's contained in a list from the first method

testAverageOne - The average grade calculated using the first method

testAverageTwo - The average grade calculated using the second method

name - The name of the person, which is contained in a tuple from the second

method

listValues - The actual grades, which are stored in a list from the first

method

gradeList - The actual grade, which are stored in a tuple from the second

method

Tasks allocated to functions:

tupleListToDict() - Changes a list of tuples into a dictionary

getSortedKeyList() - Turns the keys from a dictionary into a sorted list

computeAverage() - Computes the average grade from the list of grades

getSortedListOfTuples() - Turns a dictionary into a sorted list of tuples

"""

**Page 1 of 4**

# Functions ------------------------------------------------------------------

# Turns the given list of tuples into a dictionary where the name is the key

# and the value is a list of grades. The dictionary is then returned.

# param tupleList (list)

# returns dictionary (dict)

def tupleListToDict(tupleList):

dictionary = {}

for (name, values) in tupleList:

if dictionary.get(name) == None:

dictionary[name] = []

dictionary[name] += values

##print(dictionary)

return dictionary

# Takes a dictionary, turns the keys into a list and then sorts them. The

# list is then returned.

# param dictionary (dict)

# invoke list()

# invoke sort()

# return keyList (list)

def getSortedKeyList(dictionary):

keyList = list(dictionary.keys())

keyList.sort()

##print(keyList)

return keyList

# Takes a list of numbers and averages the numbers together. The average is

# returned.

# param numberList (list)

# invoke range()

# invoke len()

# returns average (float)

def computeAverage(numberList):

total = 0

for i in range(len(numberList)):

total += numberList[i]

##print(total)

if len(numberList) == 0:

average = 0

else:

average = total/len(numberList)

##print(average)

return average

**Page 2 of 4**

# Takes a dictionary and makes a list of tuples from the key-value pairs.

# The list is sorted, then returned.

# param dictionary (dict)

# invoke list()

# invoke sort()

# retun tupleList (list)

def getSortedListOfTuples(dictionary):

tupleList = list(dictionary.items())

tupleList.sort()

return tupleList

# Main -----------------------------------------------------------------------

# From a given, literal list of tuples with names and pairs, prints out a

# table showing the total number of scores and the average score for each

# person

def main():

# Given list of tuples

listOfTuples = [ ('Zaphod', [33, 20]), ('Zaphod', [75, 48]), ('Slartibartfast'\

,[]), ('Trillian', [98, 88]), ('Trillian', [97, 77]), ('Slartibartfast', []),

('Marvin', [2000, 500]) , ('Arthur', [42, 20]), ('Arthur', [64]),

('Trillian', [99]), ('Marvin', [450]), ('Marvin', [550]),

('Agrajag', []), ('Agrajag', []), ('Agrajag', [0]),

('Ford',[50]), ('Ford', [50]), ('Ford', [50]) ]

# Turns the given list of tuples into a dictionary where the name is the

# key and the list of grades is the value

##print(listOfTuples)

tupleDict = tupleListToDict(listOfTuples)

##print(tupleDict)

# Begins method one and sorts the keys of the dictionary in a list

dictKeys = getSortedKeyList(tupleDict)

##print(dictKeys)

# Begins method two and takes the name and the values from the dictionary

# and turns them into a list of tuples

newTupleList = getSortedListOfTuples(tupleDict)

**Page 3 of 4**

# Prints the table for method one and iterates through the sorted keys

# to make a list of numeric values.

# The list is then averaged for each name and the name, total number

# of grades and average test grade is printed.

print("Name",'\t',"Grade Total",'\t',"Average")

print("---------------------------------")

##print(getSortedListOfTuples(tupleDict))

for key in dictKeys:

listValues = tupleDict[key]

testAverageOne = computeAverage(listValues)

print('%-14s\t%-d\t%6.2f' % (key, len(listValues), testAverageOne))

# Prints the table for method two and creates a list of tuples consisting

# of names and the list of grades by iterating through the tuple list.

# Then, the variable portion of the tuple is averaged.

# Lastly, the names, number of grades, and average grade is printed

print('\n')

print("Name",'\t',"Grade Total",'\t',"Average")

print("---------------------------------")

for (names,gradeList) in newTupleList:

grades = gradeList

testAverageTwo = computeAverage(grades)

print('%-14s\t%-d\t%6.2f' % (names, len(gradeList), testAverageTwo))

main()

**Page 4 of 4**