Assignment 2 : Survey

Source Code:

#This program is running on python 3.5.0

def readData(fileName): #This converts the data from the .txt file into a list in python, doing this should speed up the process

myFile = open(fileName,'r') #opens the .txt file and enters read only more

myList = [] #list used to store contents of .txt file

for eachLine in myFile: #loop for each line in the list

depList = eachLine.split() # splits the contents of the list

department = depList[0] # stores the department as 'department'

good = depList[1] # stores the value of good reviews

fair = depList[2] #stores the value of fair reviews

poor = depList[3] #stores the value of poor reviews

header = [department,[good,fair,poor]] #header used to assign format of list

myList.append(header) #adds the line of content from the .txt file to list

myFile.close # close .txt file

return myList #returns list created

def menu(): #prints the menu for the different functions available to user

print(""" 0. Quit

1. Display menu

2. Display all survey information

3. Search department, display number of participants and average result

4. Display highest customer satisfaction

5. Display lowest customer satisfaction

6. Display departments with over 50% voted fair or poor

7. Display departments with 60% or more voted good

8. Display total participant and average value """)

def displayData(myList): #this program displays all data in the list

for departments in myList: #loop for all items in the list then prints all information on screen

print ('Department:',departments[0],' Good:',departments[1][0],' Fair:',departments[1][1],' poor:', departments[1][2])

def calculateTotal(myList): #fucntion for calculating total participants

total = 0 #total vairble used for total

for departments in myList: #creates loops for all departments in list

for result in range(3): # loop for each results in survey

total+= int(departments[1][result - 1]) #adds the results of survey to total

return total #returns total

def calculateAverage(myList): # calculate average response. formula: value of responce(1) \* weight(1) + value of responce(2) \* weight(2).../total

good = 0

fair = 0

poor = 0

for departments in myList: # loop for all departments in the list

good += int(departments[1][0]) #adds the value of good revies to total good review

fair += int(departments[1][1]) #adds the value of fair reviews to total fair reviews

poor += int(departments[1][2]) #adds the value of poor reviews to total poor reviews

average = float((good \* 3) + (fair \* 2) + (poor))/ int(calculateTotal(myList)) #calculates average reviews

return average #returns average value

def calculateTotalByDepartment(myList,depName): #calculates total review participants based on given department

valid =False #boolean varibale used to check is department exist

total = 0

for departments in myList: #loop of each department in the list

if departments[0] == depName: #if the searched department is identival to department on list

valid = True #turns valid boolean into true

for result in range(3):

total += int(departments[1][result - 1]) #adds each survey result to total

if valid: # checks if department name entered is valid, if not error message is returned, if valid total is returned

return total

else:

return 'error, department not found(check for gramatical/spelling errors)'

def calculateAverageByDepartment(myList,depName): #calculates average for a specific department entered

valid = False

for departments in myList:

if departments[0] == depName: #if the searched department is identival to department on list

valid = True

average = float(int(departments[1][0])\* 3+ int (departments[1][1])\* 2+ int(departments[1][2]))/ int(calculateTotalByDepartment(myList,depName)) #calculates average using formula: value of responce(1) \* weight(1) + value of responce(2) \* weight(2).../total

if valid: #checks if department name entered is valid, if not error message is returned, if valid average is returned

return average

else:

return 'Error calculating average(department not found)'

def displayHighest(myList): # finds department with the highest user review

highestValue = -1 # this is a imposible value to guarantee a highest reviewed department is found

for departments in myList:

currentValue = calculateAverageByDepartment(myList,departments[0]) #calculates and sets average for the department using calculateAverageByDepartment function

if float(currentValue) > highestValue: # if the current average value is greater than highest listed average value

highestValue = currentValue #sets the highestvalue to current value of department

highestDepartment = departments[0] #saves department name

return highestDepartment #returns department name with highest reiviewed

def displayLowest(myList): # finds worst performing deparment with lowest reviews

lowestValue = 4# this is a imposible value to guarantee a lowesrt reviewed department is found

for departments in myList:

currentValue = calculateAverageByDepartment(myList,departments[0]) #calculates and sets average for the department using calculateAverageByDepartment function

if float(currentValue) < lowestValue: # if current value is lower than lowest recorded value

lowestValue = currentValue #sets lowest value to curretn value

lowestDepartment = departments[0] #stores name of department with lowest average review

return lowestDepartment #returns department name

def displayPoorPerformance(myList): #this creates a list of departments with poor or fair reviews of greater than 50%

poorPerformance = [] #list for all department poor and fair reviews greater than 50%

for departments in myList:

percentage = float(int(departments[1][1]) + int(departments[1][2]))/int(calculateTotalByDepartment(myList,departments[0])) #calculates percentage of reviews that are poor and fair

if float(percentage) > .5: # if percentage is greater han 50%

poorPerformance.append(departments[0]) #adds department to poor performance department list

return poorPerformance #returns poor performance list

def displayExcellentPerformance(myList): # creates a list of best performing department based on customer reviews

excellentPerformance = [] #list for best performing departments

for departments in myList:

percentage = float(float(departments[1][0])/calculateTotalByDepartment(myList,departments[0])) #calculates the average based on no. good review/total participant for department

if float(percentage) > .6: #if percentage of good reviews is greater than 60%

excellentPerformance.append(departments[0]) #adds department to excellent performance list

return excellentPerformance #returns excellent performance list

surveyList = readData("survey.txt") #creates a list using survey.txt data

print(menu()) #prints menu on launch for the benifit of the consumer

choice = 1

while int(choice) != 0: #ends loop when user enters 0

choice = int(input("Enter your choice?(Enter the number corosponding the option)")) #allows the user to select which functions they want to access

if choice == 1: # if users chooses 1, opens menu

print(menu())

elif choice == 2: #if user chooces 2 displayData function runs

displayData(surveyList)

elif choice ==3: #if user chooces 3 calculateTotalByDepartment function is run

search = input("Enter the department?") #note, this program runs on python 3.5.0, if marking using 2.7.10 replace input into raw\_input

print ('total participants:',calculateTotalByDepartment(surveyList,search.capitalize()),".Average result:", calculateAverageByDepartment(surveyList,search.capitalize()))

elif choice ==4: #if choice = 4 runs displayHighest function

print(displayHighest(surveyList))

elif choice ==5: #if choice = 5 runs displayLowest function

print(displayLowest(surveyList))

elif choice ==6: #if choice = 6 it runs displaypoorperformance function

print (displayPoorPerformance(surveyList))

elif choice ==7: #if choice = 7 runs displayexcellentperformance function

print (displayExcellentPerformance(surveyList))

elif choice ==8: #if 8 is chosen then it runs the calculatetotal and calculateaverage then displays the message

print ('total participants:', calculateTotal(surveyList),'. Avegrage:', calculateAverage(surveyList))

elif choice == 0:

...

else: #this is a error message if the user enters a invalid value

print("Choice is invalid, try again")

print('Thank you for using our system,goodbye') #this is a goodbye message when the program is exited

**survey.txt 1**

Ladies 250 184 196

Gentlemen 167 321 459

Toys 180 150 210

Beauty 450 280 320

Technology 169 320 279

Home 120 58 45

Appliances 210 130 67

Food 180 45 89

Shoes 260 100 210

Children 179 50 80

**Survey.txt 2(used to test best performing function)]**

Ladies 250 184 196

Gentlemen 167 321 459

Toys 180 150 210

Beauty 450 280 320

Art 200 10 5

Technology 169 320 279

Home 120 58 45

Appliances 210 130 67

Sports 300 10 5

Food 180 45 89

Shoes 260 100 210

Children 179 50 80

**Testing**

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Data Used | Expected Result | Actual result |
| Text file is read correctly and data stored in memory | Survey.txt 1 | **surveyList:** [[ladies,[250,184,196]],[gentlemen,[167,321],459],[toys,[180,150,210]],[beauty,[450,280,320]],[Technology,[169,320,279]],[home,[120,58,45]],[appliance,[210,130,67]],[food,[180,45,89]],[shoes,[260,100,210]],[children’[179,50,80]]]  **Prints:**  Ladies 250 184 196  Gentlemen 167 321 459  Toys 180 150 210  Beauty 450 280 320  Technology 169 320 279  Home 120 58 45  Appliances 210 130 67  Food 180 45 89  Shoes 260 100 210  Children 179 50 80 | surveyList:  [['Ladies', ['250', '184', '196']], ['Gentlemen', ['167', '321', '459']], ['Toys', ['180', '150', '210']], ['Beauty', ['450', '280', '320']], ['Technology', ['169', '320', '279']], ['Home', ['120', '58', '45']], ['Appliances', ['210', '130', '67']], ['Food', ['180', '45', '89']], ['Shoes', ['260', '100', '210']], ['Children', ['179', '50', '80']]]  **Prints**  Department: Ladies Good: 250 Fair: 184 poor: 196  Department: Gentlemen Good: 167 Fair: 321 poor: 459  Department: Toys Good: 180 Fair: 150 poor: 210  Department: Beauty Good: 450 Fair: 280 poor: 320  Department: Technology Good: 169 Fair: 320 poor: 279  Department: Home Good: 120 Fair: 58 poor: 45  Department: Appliances Good: 210 Fair: 130 poor: 67  Department: Food Good: 180 Fair: 45 poor: 89  Department: Shoes Good: 260 Fair: 100 poor: 210  Department: Children Good: 179 Fair: 50 poor: 80 |
| Displays the entered data works correctly | Survey.txt 1 | Ladies 250 184 196  Gentlemen 167 321 459  Toys 180 150 210  Beauty 450 280 320  Technology 169 320 279  Home 120 58 45  Appliances 210 130 67  Food 180 45 89  Shoes 260 100 210  Children 179 50 80 | **Prints**  Enter your choice?(Enter the number corosponding the option)2  Department: Ladies Good: 250 Fair: 184 poor: 196  Department: Gentlemen Good: 167 Fair: 321 poor: 459  Department: Toys Good: 180 Fair: 150 poor: 210  Department: Beauty Good: 450 Fair: 280 poor: 320  Department: Art Good: 200 Fair: 10 poor: 5  Department: Technology Good: 169 Fair: 320 poor: 279  Department: Home Good: 120 Fair: 58 poor: 45  Department: Appliances Good: 210 Fair: 130 poor: 67  Department: Sports Good: 300 Fair: 10 poor: 5  Department: Food Good: 180 Fair: 45 poor: 89  Department: Shoes Good: 260 Fair: 100 poor: 210  Department: Children Good: 179 Fair: 50 poor: 80  Enter your choice?(Enter the number corosponding the option) |
| Invalid option displays error message | Enter:  111 | Error message | Enter your choice?(Enter the number corosponding the option)111  Choice is invalid, try again  Enter your choice?(Enter the number corosponding the option) |
|  | -1 | Error message | Enter your choice?(Enter the number corosponding the option)-1  Choice is invalid, try again  Enter your choice?(Enter the number corosponding the option) |
| Program stops when it should | 0 | Goodbye message | Enter your choice?(Enter the number corosponding the option)0  Thank you for using our system,goodbye |
| Display sum of value works correctly | Survey.txt 1 | Calculation: 250+ 184 +196+167+ 321+ 459 +180 +150+ 210 +450 +280+ 320 +169 +320 +279 +120 +58+ 45 +210 +130 +67+ 180+ 45 +89 +260 +100+ 210+ 179+ 50+ 80  Results  5758 | Enter your choice?(Enter the number corosponding the option)8  total participants: 5758 . Avegrage: 2.0364709968739145 |
| Display average of value works correctly | Survey.txt 1 | Calculation:(2165\*3+1638\*2+1955)/5758  result  2.036470997 | Enter your choice?(Enter the number corosponding the option)8  total participants: 5758 . Avegrage: 2.0364709968739145 |
| Displaying information about department works correctly | Survey.txt 1  Enters:  Gentlemen | Gentlemen  Cal:167+321+459  Total: 947  Cal: (167\*3+321\*2+459)/947  Average: 1.691.. | Enter your choice?(Enter the number corosponding the option)3  Enter the department?Gentlemen  total participants: 947 .Average result: 1.6916578669482576 |
|  | Enter:  Home | Home  Total:223  Average:2.336 | Enter the department?Home  total participants: 223 .Average result: 2.336322869955157 |
|  | toys | Toys  Total:540  Avergae:1.944 | Enter the department?toys  total participants: 540 .Average result: 1.9444444444444444 |
|  | shoes | Shoes  Total:570  Average:2.0877 | Enter your choice?(Enter the number corosponding the option)3  Enter the department?shoes  total participants: 570 .Average result: 2.087719298245614 |
| If information does not exit and error message | sports | Error message | sports  total participants: error, department not found(check for gramatical/spelling errors) .Average result: Error calculating average(department not found)  Enter your choice?(Enter the number corosponding the option) |
| Displays highest rated department works correctly | Survey.txt1 | Ladies 2.085714285714286  Gentlemen 1.6916578669482576  Toys 1.9444444444444444  Beauty 2.123809523809524  Technology 1.8567708333333333  Home 2.336322869955157  Appliances 2.3513513513513513 \*highest val  Food 2.289808917197452  Shoes 2.087719298245614  Children 2.320388349514563  Answer:  Appliances | Enter your choice?(Enter the number corosponding the option)4  Appliances |
| Displays lowest rated department works correctly | Survey.txt1 | Ladies 2.085714285714286  Gentlemen 1.6916578669482576 \*lowest val  Toys 1.9444444444444444  Beauty 2.123809523809524  Technology 1.8567708333333333  Home 2.336322869955157  Appliances 2.3513513513513513  Food 2.289808917197452  Shoes 2.087719298245614  Children 2.320388349514563  Answer:  Gentlemen | Enter your choice?(Enter the number corosponding the option)5  Gentlemen |
| Displaying below average performance departments work correctly | Survey.txt1 | Formula:  Ladies :  (184+196)/(184+196+250)  Results(if over .5(50%))  Ladies 0.6031746031746031  Gentlemen 0.8236536430834214  Toys 0.6666666666666666  Beauty 0.5714285714285714  Technology 0.7799479166666666  Home 0.4618834080717489  Appliances 0.48402948402948404  Food 0.4267515923566879  Shoes 0.543859649122807  Children 0.42071197411003236  Answer:  Ladies,gentlemen,toys,beuty,technology,shoes | Enter your choice?(Enter the number corosponding the option)6  ['Ladies', 'Gentlemen', 'Toys', 'Beauty', 'Technology', 'Shoes'] |
| Displays above average performance works correctly | Survey.txt 1 | Formula:  Ladies: 250/184+196+250  Results:(if over .6(60%))  Ladies 0.3968253968253968  Gentlemen 0.17634635691657866  Toys 0.3333333333333333  Beauty 0.42857142857142855  Technology 0.22005208333333334  Home 0.5381165919282511  Appliances 0.515970515970516  Food 0.5732484076433121  Shoes 0.45614035087719296  Children 0.5792880258899676  Answer:  null | Enter your choice?(Enter the number corosponding the option)7  [] |
|  | Survey.txt 2 | Result:(if over .6(60%))  Ladies 0.3968253968253968  Gentlemen 0.17634635691657866  Toys 0.3333333333333333  Beauty 0.42857142857142855  Art 0.9302325581395349  Technology 0.22005208333333334  Home 0.5381165919282511  Appliances 0.515970515970516  Sports 0.9523809523809523  Food 0.5732484076433121  Shoes 0.45614035087719296  Children 0.5792880258899676  Answer  Art, Sports | Enter your choice?(Enter the number corosponding the option)7  ['Art', 'Sports'] |