```python
#
# Machine Problem 4
# CSCI 1300
# Bruce Miller
#
# Description:
# Get readable list of student scores
# Add test average to each student in list
# Calculate  total averages
# Print list neatly
# Reorder studentScores by Name and Average, and print each list neatly
#

def getScores():
    #
    # Opens the data file of names and scores... firstName, lastName, score1,
    # score2, score3... reads each line of data as a str, divides the line into
    # the 5 values... str, str, int, int, int... puts those values in a list,
    # and returns a list of those lists.
    #
    # There are no parameters.
    #
    # Returns a list of lists... each list contains a str, str, int, int, int.
    #
    file = open("C:\\Users\\Bruce\\Desktop\\Python Coding\\Class Projects\\1300 -
MP4 Data.txt")
    lines = file.readlines()
    lines[len(lines)-1] += "\n" #last character in last line is not "\n" (FIX)
    classInfo = [[] for i in range(len(lines))]
    string = ""
    for i, currentLine in enumerate(lines):
        for character in currentLine:
            if character not in (" ", "\n"):
                string += character
            else:
                try:
                    classInfo[i].append(int(string))
                except ValueError:
                    classInfo[i].append(string)
                string = ""
    return classInfo


def addTestAverage(studentScores):
    #
    # Finds the average of each student's test scores, and then appends that
    # average onto the end of that student's list. So, each student list now
    # contains str, str, int, int, int, float.
    #
    # studentScores A list of lists, each list contains a str, str, int,
```

```python
    # int, int which are firstName, lastName, test1, test2,
    # test3.
    #
    # There is no return value.
    #
    for i, student in enumerate(studentScores):
        sumScore = 0
        for score in student[2:]:
            sumScore += int(score)
        studentScores[i].append(sumScore/3)


def calcTotals(studentScores):
    #
    # Finds the average of test1, test2, test3, and the total average. Returns
    # those 4 values in a list.
    #
    # studentScores A list of lists, each list contains a str, str, int,
    # int, int, float which are firstName, lastName, test1,
    # test2, test3, average.
    #
    # Returns a list with 4 values... float, float, float, float... which are
    # test1 avg, test2 avg, test3 avg, total avg.
    #
    totals = [0, 0, 0, 0]
    for student in studentScores:
        for i in range(4):
            totals[i] += student[i + 2]
    averages = [x / len(studentScores) for x in totals]
    return averages


def printScores(studentScores, totals):
    #
    # Prints out the entire list including firstName, lastName, score1, score2,
    # score3, average. There is a header for each column. The totals are
    # printed at the end.
    #
    # studentScores A list of lists, each list contains a str, str, int,
    # int, int, float which are firstName, lastName, test1,
    # test2, test3, average.
    # totals A list of 4 float values... the averages for test1,
    # test2, test3, and totalAverage.
    #
    # There is no return value.
    #
    print("Name                       Exam1  Exam2  Exam3    Avg")

    for student in studentScores:
        print(f"{student[0]+" "+student[1]:<20}", end="")
```

```python
        print(f"{student[2]:>7}{student[3]:>7}{student[4]:>7}", end="")
        print(f"{round(student[5],2):>7.2f}")

    print(f"Total               ", end="")
    for t in totals:
        print(f"{round(t, 2):>7}", end="")
    print()
    print()


def sortByName(studentScores):
    #
    # Sorts the list of student info by the student's last name. Uses the
    # Bubble Sort algorithm.
    #
    # studentScores A list of lists, each list contains a str, str, int,
    # int, int, float which are firstName, lastName, test1,
    # test2, test3, average.
    #
    # There is no return value.
    #
    for i in range(len(studentScores) - 1):
        for j in range(len(studentScores) - 1):
            if studentScores[j][1] > studentScores[j+1][1]:
                temp = studentScores[j]
                studentScores[j] = studentScores[j+1]
                studentScores[j+1] = temp


def sortByAverage(studentScores):

    #
    # Sorts the list of student info by the test average. Uses the
    # Bubble Sort algorithm.
    #
    # studentScores A list of lists, each list contains a str, str, int,
    # int, int, float which are firstName, lastName, test1,
    # test2, test3, average.
    #
    # There is no return value.
    #
    for i in range(len(studentScores) - 1):
        for j in range(len(studentScores) - 1):
            if studentScores[j][5] < studentScores[j+1][5]:
                temp = studentScores[j]
                studentScores[j] = studentScores[j+1]
                studentScores[j+1] = temp


studentScores = getScores()
```

```
addTestAverage(studentScores)
totals = calcTotals(studentScores)
printScores(studentScores, totals)

sortByName(studentScores)
printScores(studentScores, totals)

sortByAverage(studentScores)
printScores(studentScores, totals)
```