machine_problems/mp4.py

```
#
# Machine Problem 4
# Prince Awuah Karikari
#
# Description: This script reads data from a file, parse the data
#              and computes the average of each student, each course
#              and the total average. We print the original data from the
#              file, sort it by student's last name and student's average
#              and print the results to the output


def getScores():

    #
    # Opens the data file of names and scores... firstName, lastName, score1,
    # score2, score3, score4... reads each line of data as str, divides the
    # line into the 6 values... str, str, int, 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, int.
    #

    results = []

    with open("5010 - MP4 Data.txt") as data:
        for line in data:
            firstName, lastName, score1, score2, score3, score4 = line.split()
            results.append(
                [
                    firstName,
                    lastName,
                    int(score1),
                    int(score2),
                    int(score3),
                    int(score4),
                ]
            )

    return results


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, int, float.
    #
    # studentScores     A list of lists, each list contains a str, str, int,
```

```
    #                   int, int, int which are firstName, lastName, test1,
    #                   test2, test3, test4.
    #
    # There is no return value.
    #
    for data in studentScores:
        data.append(sum(data[2:]) / len(data[2:]))


def calcTotals(studentScores):

    #
    # Finds the average of test1, test2, test3, test4, and total average.
    # Returns those 5 values in a list.
    #
    # studentScores     A list of lists, each list contains a str, str, int,
    #                   int, int, int, float which are firstName, lastName,
    #                   test1, test2, test3, test4, average.
    #
    # Returns a list with 5 values... float, float, float, float, float...
    # which are test1 avg, test2 avg, test3 avg, test4 avg, total avg.
    #
    testTotals = [0, 0, 0, 0]

    for data in studentScores:
        for i in range(len(testTotals)):
            testTotals[i] += data[i + 2]

    avgs = [total / len(studentScores) for total in testTotals]
    overallAvg = sum(avgs) / len(testTotals)

    return avgs + [overallAvg]


def printScores(studentScores, totals):

    #
    # Prints out the entire list including firstName, lastName, score1, score2,
    # score3, score4, 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, int, float which are firstName, lastName, test1,
    #                   test2, test3, test4, average.
    # totals            A list of 5 float values... the average for test1,
    #                   test2, test3, test4, and totalAverage.
    #
    # There is no return value.
    print(
        f"\n{'Name':22} {'Exam1':>6} {'Exam2':>6} {'Exam3':>6} {'Exam4':>6}
{'Avg':>6}"
    )

    for (
```

```
        firstName,
        lastName,
        score1,
        score2,
        score3,
        score4,
        studentAvg,
    ) in studentScores:
        fullName = f"{firstName} {lastName}"
        print(
            f"{fullName:22} {score1:>6} {score2:>6} {score3:>6} {score4:>6} {studentAvg:>6.2f}"
        )

    print(
        f"{'Total':21} {totals[0]:7.2f} {totals[1]:>6.2f} {totals[2]:>6.2f} {totals[3]:6.2f} {totals[4]:6.2f}"
    )


def sortByName(studentScores):

    #
    # Sorts the list of student info by the student's last name. Uses the
    # Bubble algorithm.
    #
    # studentScores     A list of lists, each list contains a str, str, int,
    #                   int, int, int, float which are firstName, lastName, test1,
    #                   test2, test3, test4, 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, int, float which are firstName, lastName, test1,
    #                   test2, test3, test4, average.
    # There are not return value.
    #
    for i in range(len(studentScores) - 1):
        for j in range(len(studentScores) - 1):
```

```
            if studentScores[j][6] < studentScores[j + 1][6]:
                temp = studentScores[j]
                studentScores[j] = studentScores[j + 1]
                studentScores[j + 1] = temp


if __name__ == "__main__":
    scores = getScores()
    addTestAverage(scores)
    printScores(scores, calcTotals(scores))
    sortByName(scores)
    printScores(scores, calcTotals(scores))
    sortByAverage(scores)
    printScores(scores, calcTotals(scores))

```