

machine_problems/mp4.py

```
1 #
2 # Machine Problem 4
3 # Prince Awuah Karikari
4 #
5 # Description: This script reads data from a file, parse the data
6 #               and computes the average of each student, each course
7 #               and the total average. We print the original data from the
8 #               file, sort it by student's last name and student's average
9 #               and print the results to the output
10
11
12 def getScores():
13
14     #
15     # Opens the data file of names and scores... firstName, lastName, score1,
16     # score2, score3, score4... reads each line of data as str, divides the
17     # line into the 6 values... str, str, int, int, int, int... puts those values
18     # in a list, and returns a list of those lists.
19     #
20     # There are no parameters.
21     #
22     # Returns a list of lists... each list contains a str, str, int, int, int, int.
23     #
24
25     results = []
26
27     with open("5010 - MP4 Data.txt") as data:
28         for line in data:
29             firstName, lastName, score1, score2, score3, score4 = line.split()
30             results.append(
31                 [
32                     firstName,
33                     lastName,
34                     int(score1),
35                     int(score2),
36                     int(score3),
37                     int(score4),
38                 ]
39             )
40
41     return results
42
43
44 def addTestAverage(studentScores):
45
46     #
47     # Finds the average of each student's test scores, and then appends that
48     # average onto the end of that student's list. So, each student list now
49     # contains str, str, int, int, int, int, float.
50     #
51     # studentScores      A list of lists, each list contains a str, str, int,
```

```
52         #             int, int, int which are firstName, lastName, test1,
53         #             test2, test3, test4.
54         #
55         # There is no return value.
56         #
57         for data in studentScores:
58             data.append(sum(data[2:]) / len(data[2:]))
59
60
61 def calcTotals(studentScores):
62
63     #
64     # Finds the average of test1, test2, test3, test4, and total average.
65     # Returns those 5 values in a list.
66     #
67     # studentScores      A list of lists, each list contains a str, str, int,
68     #                     int, int, int, float which are firstName, lastName,
69     #                     test1, test2, test3, test4, average.
70     #
71     # Returns a list with 5 values... float, float, float, float, float...
72     # which are test1 avg, test2 avg, test3 avg, test4 avg, total avg.
73     #
74     testTotals = [0, 0, 0, 0]
75
76     for data in studentScores:
77         for i in range(len(testTotals)):
78             testTotals[i] += data[i + 2]
79
80     avgs = [total / len(studentScores) for total in testTotals]
81     overallAvg = sum(avgs) / len(testTotals)
82
83     return avgs + [overallAvg]
84
85
86 def printScores(studentScores, totals):
87
88     #
89     # Prints out the entire list including firstName, lastName, score1, score2,
90     # score3, score4, average. There is a header for each column. The totals are
91     # printed at the end.
92     #
93     # studentScores      A list of lists, each list contains a str, str, int,
94     #                     int, int, int, float which are firstName, lastName, test1,
95     #                     test2, test3, test4, average.
96     # totals              A list of 5 float values... the average for test1,
97     #                     test2, test3, test4, and totalAverage.
98     #
99     # There is no return value.
100    print(
101        f"\n{'Name':22} {'Exam1':>6} {'Exam2':>6} {'Exam3':>6} {'Exam4':>6}
{'Avg':>6}"
102    )
103
104    for (
```

```
105     firstName,
106     lastName,
107     score1,
108     score2,
109     score3,
110     score4,
111     studentAvg,
112 ) in studentScores:
113     fullName = f"{firstName} {lastName}"
114     print(
115         f"{fullName:22} {score1:>6} {score2:>6} {score3:>6} {score4:>6}
{studentAvg:>6.2f}"
116     )
117
118     print(
119         f"{'Total':21} {totals[0]:7.2f} {totals[1]:>6.2f} {totals[2]:>6.2f}
{totals[3]:6.2f} {totals[4]:6.2f}"
120     )
121
122
123 def sortByName(studentScores):
124
125     #
126     # Sorts the list of student info by the student's last name. Uses the
127     # Bubble algorithm.
128     #
129     # studentScores      A list of lists, each list contains a str, str, int,
130     #                    int, int, int, float which are firstName, lastName, test1,
131     #                    test2, test3, test4, average.
132     #
133     # There is no return value.
134     #
135
136     for i in range(len(studentScores) - 1):
137         for j in range(len(studentScores) - 1):
138             if studentScores[j][1] > studentScores[j + 1][1]:
139                 temp = studentScores[j]
140                 studentScores[j] = studentScores[j + 1]
141                 studentScores[j + 1] = temp
142
143
144 def sortByAverage(studentScores):
145
146     #
147     # Sorts the list of student info by the test average. Uses the
148     # Bubble Sort algorithm
149     #
150     # studentScores      A list of lists, each list contains a str, str, int,
151     #                    int, int, int, float which are firstName, lastName, test1,
152     #                    test2, test3, test4, average.
153     # There are not return value.
154     #
155     for i in range(len(studentScores) - 1):
156         for j in range(len(studentScores) - 1):
```

```
157         if studentScores[j][6] < studentScores[j + 1][6]:
158             temp = studentScores[j]
159             studentScores[j] = studentScores[j + 1]
160             studentScores[j + 1] = temp
161
162
163 if __name__ == "__main__":
164     scores = getScores()
165     addTestAverage(scores)
166     printScores(scores, calcTotals(scores))
167     sortByName(scores)
168     printScores(scores, calcTotals(scores))
169     sortByAverage(scores)
170     printScores(scores, calcTotals(scores))
171
```