

```

1 """
2 temperatures_two_dim.py
3 @author ITP 150 Student
4 Created Current Date
5 This script stores information about temperatures recording the dates,
6 lows, highs, and calculating the average temperature for the date in
7 a two dimensional list.
8 """
9
10
11 # num_days will set the sizes of the lists
12 num_days = int(input('How many days do you want to track temperatures?'))
13 COLS = 4
14 sum_high_temps = 0
15 HIGH_TEMP = 90
16
17 # initializing the lists by datatype and the size via num_days
18 temperatures = [[0 for col in range(COLS)] for row in range(num_days)]
19 print(temperatures) # notice the [[0, 0, 0, 0], ...] to indicate 2 dim list
20
21
22 # loop that loads the values into the dates, lows, highs columns [0, 1, 2]
23 # within the temperatures list and calculates the values for the
24 # average temps column which is column 3.
25 for row in range(0, num_days, 1):
26     temperatures[row][0] = input('Please enter the date (ex. Jul. 25):')
27     print('Please enter the low temperature for ', temperatures[row][0], ':')
28     temperatures[row][1] = float(input())
29     print('Please enter the high temperature for ', temperatures[row][0], ':')
30     temperatures[row][2] = float(input())
31     temperatures[row][3] = (temperatures[row][1] + temperatures[row][2]) / 2
32 print(temperatures) # print to see the data then comment or delete
33
34 print('-' * 50)
35 # loop that prints the details more prettily from the lists
36 print(f'{"Dates":10} {"Lows":>10} {"Highs":>10} {"Average Temps":>15}')
37 for row in range(0, num_days, 1):
38     print(f'{temperatures[row][0]:<10s} {temperatures[row][1]:>10.1f} \
39 {temperatures[row][2]:>10.1f} {temperatures[row][3]:>10.1f}')
40 print('-' * 50)
41
42 # algorithm to find the lowest low and its date
43 lowest_low = temperatures[0][1] # use min() only with single dim list
44 lowest_low_date = temperatures[0][0]
45
46 for row in range(0, num_days, 1):
47     if temperatures[row][1] < lowest_low:
48         lowest_low = temperatures[row][1]
49         lowest_low_date = temperatures[row][0]
50
51 print(f'{"Lowest Low":30s} {lowest_low:>12.1f}')
52 print(f'{"Lowest Low Date":30s} {lowest_low_date:>12s}')
53
54 # algorithm to find the highest high and its date
55 highest_high = temperatures[0][2]
56 highest_high_date = temperatures[0][0]
57
58 for row in range(0, num_days, 1):

```

```

59     if temperatures[row][2] > highest_high:
60         highest_high = temperatures[row][2]
61         highest_high_date = temperatures[row][0]
62
63 print(f'{"Highest High":30s} {highest_high:>12.1f}')
64 print(f'{"Highest High Date":30s} {highest_high_date:>12s}')
65
66 # code to calculate the average of the highs
67 for row in range(0, num_days, 1):
68     sum_high_temps = sum_high_temps + temperatures[row][2]
69
70 average_highs = sum_high_temps / len(temperatures) # use sum() for a single dim
list
71 print(f'{"Average Highs":30s} {average_highs:>12.1f}')
72
73 # algorithm to print each day where high >= 90 which is value
74 # stored in HIGH_TEMP. When entering test data, enter at least 1
75 # day with an 90 as the high
76 for row in range(0, num_days, 1):
77     if temperatures[row][2] >= HIGH_TEMP:
78         print('The average temp exceeded', HIGH_TEMP, 'on',
79               temperatures[row][0], 'recorded at', temperatures[row][2], '.')
80 print('-' * 50)
81

```