G12-Term1-Test1

October 10, 2019

1 The Dragon Academy 2019-2020

2 G11/12 Computer Science Term 1 Test 1

2.0.1 Instructions

This python notebook can be downloaded from here
Make sure to provide your name in the next cell.

Submission: Once finished, save the notebook as PDF and send as an attachment by email both the notebook itself as well as its PDF version. Please, use as subject of the email "Term 1 Test 1" There is a quiz and a problem section. The first weighs 40% of the total mark of this test and is common to both G11 and G12 students. The problem section consists as well in 2 exercises, the first one being common for all and the second being specific to your grade.

Each quiz question weighs the same, and so do each problem.

2.0.2 PROVIDE YOUR ANSWERS IN THIS NOTEBOOK

3 NAME:

4 QUIZ

1. (KticA) A dictionary is a collection which is unordered, changeable and indexed. In Python dictionaries are written with curly brackets, and they have keys and values. Example:

```
thisIsADictionary = {'key1': 367, 'key2': 'value2', 'c': [3,5], 0: {'a':1, 'b':3}}
print(thisIsADictionary['key1'])   #prints 367
print(thisIsADictionary[0])        #prints {'a': 1, 'b': 3}
```

1. What does the following loop print `for w in thisIsADictionary: print(w)`?
2. Idem for `for k,v in thisIsADictionary.items(): print(k,v)`
3. Write down the types of every key and its corresponding value.
4. Explain what the following code does to this dictionary. Does it work?: `thisIsADictionary['key']='oh!'`

5. Explain what the following code does: `thisIsADictionary[0]='val'`

6. (kTIca) In this following code, all three variables `d1, d2, d3` are copies of the original dictionary. However, they are not equal. Using the methods learned from the previous question, experiment with these three copying methods and explain which one is the *best/most correct*, i.e., which one creates a new copy that is more faithful to what we would understand as *independent copy*.

```
import copy
d1 = thisIsADictionary.copy()
d2 = dict(thisIsADictionary)
d3 = copy.deepcopy(thisIsADictionary)
```

5 Problems

5.1 1

Write a function `csvRead` the reads a file, say 'table.csv', containing a comma-separated list of values and outputs an array of dictionaries each corresponding to each row of the file after the first row. The very first row of the file contains the labels of each column. You can treat these labels and all values as strings.

Sample Input

brand, horsepower, L/100Km highway, price bmw, 150, 6.5, 50000 mercedes, 160, 6.4, 60000

Sample Output

```
[
{'brand': 'bmw', 'horsepower': '150', 'L/100Km': '6.5', 'price': '50000' },
{'brand': 'mercedes', 'horsepower': '160', 'L/100Km': '6.4', 'price': '60000' }
]
```

5.2 2 (G11 only)

High Tide
Input file: probd.in
Output file: probd.out
A planet has n moons revolving about it in constant clockwise coplanar circular orbits. How often do all the moons appear directly overhead as viewed from some point on the planet? We will call such a situation a 'vertical alignment'.

Your input consists of a number of sets of lines; each set consists of an integer n, indicating the number of moons, followed by n distinct positive integers, one per line, indicating the exact period of revolution for each moon, in days. (Thus there are n+1 lines of data for each set with the first line containing n.) The last line contains only a zero.

For each input set, except the last, generate a line of output indicating the interval in days, to two decimal places, between consecutive vertical alignments.

Sample Input

2
20

30
3
20
30
40
2
10
3
0
Sample Output
60.00
120.00
4.29

5.3 3 (G12 only)

Alien Invasion
Input file: probb.in
Output file: probb.out
Earth is being invaded by space aliens. Earth defence forces have rallied a number of anti-spacecraft guns. However, they have a bug in their aiming hardware: initially they are aimed straight up, and this aim can only be adjusted downward.

Thousands of alien craft are streaking towards Earth as we speak – and yes, some of them are even aimed at Canada. The Earth defence forces must now come into play. Each gun can fire as many shots as necessary, and can be re-fired as often and as quickly as necessary, but only to a lower setting. Thus if a spacecraft came in at height 3 and then another at height 2, one gun could eliminate both, but could not if they came in the other order. The Earth has only a finite number of guns and it is unknown how many alien craft are coming in. Thus they need a way to minimize the number of guns for a given set of incoming alien craft. Guess what? This is where you come in!

Input
The data will consist of several sets of data. The first line of each set will contain one positive integer n (n < 100000), where n is the number of incoming alien craft. The next n lines will contain one floating point number giving the heights of the incoming alien craft in order of arrival (ie, the order the guns must eliminate them). The last line in the data file will contain only zero, ie n = 0.

Output For each set other than the final, n = 0, case, one integer specifying the minimum number of guns required to eliminate EVERY alien craft is to be output.

Sample Input
10
4.0
2.0
3.0
4.0
5.0
3.0
1.0
4.0
2.0

5.0
0

Output for sample input

4

[ ]: