BIOS 6642 – Take-Home Assignment Three

Due at 11:59PM (Mountain Time), Tuesday, April 13, 2021

Please submit your source codes (.py or .ipynb file) and screenshots of the output of your codes. Your codes should be properly documented or commented.

- Q1 (30%). Please write a function that takes a tuple of integers, e.g., ***tup***, as input. This input tuple, ***tup,*** provides daily average temperatures, with each element for one day. The elements with lower indexes in the tuple correspond to dates earlier than those with higher indexes. For each day in the input tuple, the function needs to count as least how many days it needs to get a higher temperature and store this value (i.e., days to get a higher temperature) in a list, e.g., ***lst,*** which should have the same length as the input tuple. If there is no higher temperature in the future, set the corresponding value to 0 in ***lst***. The function returns this list. Assume the length of input tuple can be any number between 1 and 100 (including 1 and 100).

  For example,

  - If ***tup*** = *(20, 15, 30, 66, 36, 36, 51, 89, 92)*, then ***lst*** should be *[2, 1, 1, 4, 2, 1, 1, 1, 0].*
  - If ***tup*** = *(29, 30, 21, 28, 9, 12)*, then ***lst*** should be *[1, 0, 1, 0, 1, 0].*

  In addition to the function, please write some code to test the function.

- Q2 (35%). Please write a function that takes as input two positive integers, ***num*** and ***n***. The number of digits of ***num*** is not smaller than the value of ***n***. The function needs to delete ***n*** digits from ***num*** such that the resulting, new number is the smallest. The function returns the new number; if all the digits are deleted, the function returns *0*.

  For example,

  - If ***num*** = *2030* and ***n*** = *1*, then the function should return *30*, because deleting the leading *2* produces a number *30*, which is the smallest.
  - If ***num*** = *3712136* and ***n*** = *3*, then the function should return *1136*, because deleting digits *3 (the first one), 7* and *2* produces the smallest number, *1136*.
  - If ***num*** = *19* and ***n*** = *2*, then the function should return *0*, because all the digits are deleted.

  In addition to the function, please write some code to test the function.

- Q3 (35%). There is a JSON file, ***olympic_medals.json***, which contains the number of Summer Olympic medals for 12 Olympic committees or countries. The country names are in the array associated with the "key", *Country*. For each country, the numbers of gold, silver and bronze medals are stored in corresponding elements of the array associated with the "key", *Medals*, as shown in Figure 1. Data source: https://en.wikipedia.org/wiki/All-time_Olympic_Games_medal_table

Figure 1. Numbers of Summer Olympic medals for 12 countries.

Please download the JSON file onto your computer and do not modify the content of the file. Please write a program to read data from the file and

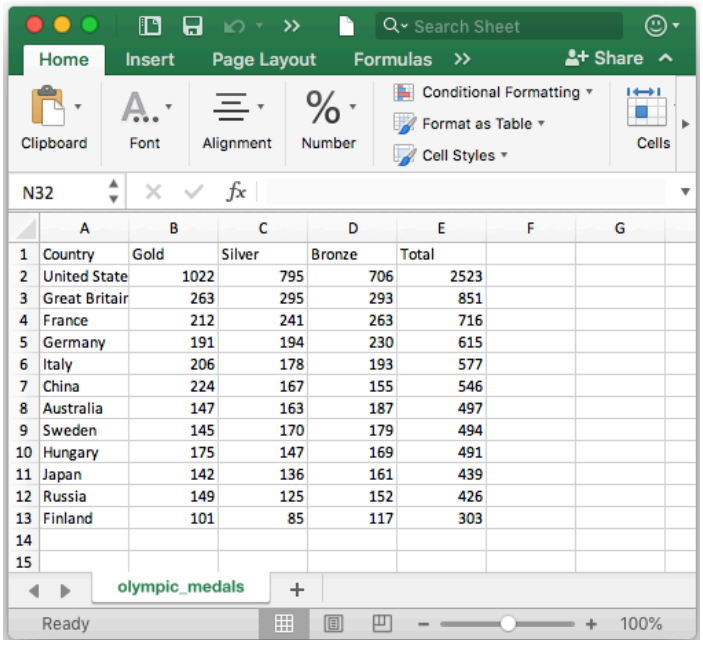
- Print the data within the JSON file on the screen in a tabular format: columns 1, 2, 3, and 4 represent country, gold, silver and bronze medal numbers, respectively. Please print the data in decreasing order in terms of the number of gold medals. An example printing output is shown in Figure 2.

Country	Gold	Silver	Bronze
United States	1022	795	706
Great Britain	263	295	293
China	224	167	155
France	212	241	263
Italy	206	178	193
Germany	191	194	230
Hungary	175	147	169
Russia	149	125	152
Australia	147	163	187
Sweden	145	170	179
Japan	142	136	161
Finland	101	85	117

Figure 2. An example printing output.

- Calculate the total number of medals (including gold, silver and bronze) for each country and then write the country names and their corresponding

numbers of gold, silver, bronze and total medals into a CSV file. Please write the data in decreasing order in terms of the number of total medals. An example output CSV file is shown in Figure 3.



	A	B	C	D	E
1	Country	Gold	Silver	Bronze	Total
2	United State	1022	795	706	2523
3	Great Britair	263	295	293	851
4	France	212	241	263	716
5	Germany	191	194	230	615
6	Italy	206	178	193	577
7	China	224	167	155	546
8	Australia	147	163	187	497
9	Sweden	145	170	179	494
10	Hungary	175	147	169	491
11	Japan	142	136	161	439
12	Russia	149	125	152	426
13	Finland	101	85	117	303
14					
15					

Figure 3. An example output CSV file.

For this question, please submit your source codes (.py or .ipynb file), screenshots of the printing of your codes and the output CSV file.