

Q1 Welcome!

0 Points

Welcome to Part 2 of the Final Exam for CSE 160 22au! Have fun!

Don't forget to also submit Part 1 (function writing)!

Note: Please be careful when writing code snippets that require indentation. If writing code directly in the box, use four spaces to indicate an indentation. Alternatively, you may wish to write your code in VSCode and copy and paste it into the text box provided. Regardless of how you enter your code, please return to the question and confirm that indentation is as you would like it.

Point values on Part 1 and Part 2 are not final and may be adjusted slightly.

(Optional - Just for fun!) If you were designing your own programming language, what animal would YOU name it after and why?

Q2 File I/O

6 Points

Welcome to the 2022 World Cup! You are working as an analyst for FIFA and have been provided with a file named `goals.txt` in which each row contains the name of a country followed by one or more integers representing the number of goals that country has scored in a game, with a single space coming before each number. Note: A country could appear more than once in the file. You may assume the file consist only of lines that have been formatted correctly.

You have been asked to write python code that will calculate the **lowest number of goals** that each country in this file has scored. Fill in the code below to create a dictionary named `goals_dict` which maps each country name to the lowest number of goals it scored in a game.

For example, if the contents of the file `goals.txt` was:

```
Ecuador 4 3 11 4
USA 1 0
Argentina 2 10 6
Senegal 0 3
Argentina 7 0 9
```

Then `goals_dict` dictionary would contain:

```
{"Ecuador": 3, "USA": 0, "Senegal": 0, "Argentina": 0}
```

Write your code assuming the following three lines exist:

```
goals_dict = {}
goals_file = "goals.txt"

# Your Code Here

print(goals_dict)
```

Write your python code here. You do not need to write a function. Use 4 spaces to indicate an indentation if typing directly into the textbox, or copy and paste from VSCode. Regardless, please be sure that your indentation is as you would like it.

```
input_file = open("goals.txt")
# Process the data for one line of the file
# note: this could be the second time you see this country
for line in input_file:
    line_lst = line.split()
    country = line_lst[0]
    if country in goals_dict: # to handle country on > 1 line
        min_score = goals_dict[country]
    else:
        min_score = None
    for score in line_lst[1:]:
        if min_score is None:
            min_score = int(score)
        elif int(score) < min_score:
            min_score = int(score)
# Can reset a previous mapping in the dict
```

```
goals_dict[country] = min_score

input_file.close()
```

Q3 Graphs

6 Points

You're working on HW5 Part 2, and are building up your facebook graph. But oops! You accidentally added every user id as a string to your graph with the characters "fb" at the end. For example user 367 was added as the string "367fb".

You decide to write a function `clean_graph(input_graph)` to fix this problem. This function takes in `input_graph`, a Graph from `networkx` as in homework 5, with nodes named as strings of the form "`idfb`", where `id` is the integer ID of the user. The function should return a graph that represents the same edges and nodes as `input_graph`, but the nodes are now named as integers corresponding to `id`, instead of strings. The `id` length is variable (may be one or more digits long). You should only use parts of `networkx` we used in class. This is the code you have so far:

```
def clean_graph(input_graph):
    """
    Returns a graph corresponding to input_graph that contains
    the same edges and nodes, but where all nodes are integers
    and trailing characters "fb" have been removed.
    """
    cleaned_graph = nx.Graph()

    # Your code here

    return cleaned_graph
```

Write your code here:

```
# For each edge in input_graph, create a new edge in cleaned_graph
# that has the nodes named appropriately.
for edge in input_graph.edges():
    new_node1 = int(edge[0][:-2])
```

```
new_node2 = int(edge[1][:-2])
cleaned_graph.add_edge(new_node1, new_node2)
```

Q4 Sorting

4 Points

Suppose you have the following:

```
fall_enroll = [('CSE 160', 'Ruth Anderson', 180),
               ('CSE 446', 'Ruth Anderson', 180),
               ('CSE 446', 'Brett Wortzman', 200),
               ('CSE 446', 'Brett Wortzman', 180)]
```

where each item in `fall_enroll` consists of a tuple containing (class name, instructor, class size).

Write code to create a sorted version of `fall_enroll` in the variable `output_list`. **You should NOT modify `fall_enroll`.**

The list should be sorted alphabetically by class name. If there is a tie between classes, sort by number of students where the greatest number of students comes first. Finally, if there is a tie between class name and number of students, sort by professor name alphabetically.

You should NOT use any loops and you should NOT use a loop

comprehension in your code. Your code should work for all non-empty properly-formed lists, not just this one. You do not need to write a function. Don't forget to import anything you need to solve this problem. After your code runs, `output_list` should be:

```
output_list = [('CSE 160', 'Ruth Anderson', 180),
               ('CSE 446', 'Brett Wortzman', 200),
               ('CSE 446', 'Brett Wortzman', 180),
               ('CSE 446', 'Ruth Anderson', 180)]
```

Write your code here:

```
from operator import itemgetter
by_prof = sorted(fall_enroll, key=itemgetter(1))
```

```
by_size = sorted(by_prof, key=itemgetter(2), reverse=True)
output_list = sorted(by_size, key=itemgetter(0))
```

Q5 Style

4 Points

Your friend has written this snippet of code to calculate the average height of everyone in the class and print it out, but the variable names are terrible. Help your friend fix the code by coming up with better variable names:

```
d = {"Sally": 54, "Mark": 60, "Bob": 59, "Anna": 61}

s = 0

for x in d:
    s += d[x]

a = s / len(d)

print(a)
```

Give a better variable name for each of these variables:

d:

heights_dict

s:

sum_of_heights

x:

person

a:

average_height

Q6 Classes

9 Points

You've been hired to work at a new manga store in the International District! You are responsible for managing the store's manga sales, and you are given this python class to use:

```
class MangaManager:
    """A class to manage inventory and funds of a Manga store."""
    def __init__(self, location, prices_data, initial_funds):
        """ Creates a new MangaManager
            location (string): location of the store
            prices_data (dictionary): maps a manga (string)
                to its price in dollars (integer)
            initial_funds (integer): dollars the store starts out with
        """
        self.store_location = location
        self.manga_stock = {}
        self.manga_prices = prices_data
        self.funds = initial_funds

    def add_manga(self, manga_name, amount):
        """ Adds a number of manga to the store's stock.
            manga_name (string): name of manga we are adding
            amount (integer): number of this manga we are adding
        """
        if manga_name in self.manga_stock:
            self.manga_stock[manga_name] += amount
        else:
            self.manga_stock[manga_name] = amount

    def sell_manga(self, manga_name, amount):
        """
        Sells a number of a specific manga, removing from the store's stock,
        increasing the store's funds, and printing a message. If the store
        does not have enough copies of the manga, prints a message stating so.
            manga_name (string): name of manga we are selling
            amount (integer): number of this manga we are selling
        """
        if (manga_name not in self.manga_stock or
            self.manga_stock[manga_name] < amount):
            print("Not enough copies of", manga_name, "in stock!")
        else:
            revenue = self.manga_prices[manga_name] * amount
            self.manga_stock[manga_name] -= amount
            self.funds += revenue
            print("Sold", amount, "copies of", manga_name)
```

Q6.1 kino_seattle

2 Points

Write client code to create a new `MangaManager` object for the store, assigning it to a variable called `kinokuniya_seattle`. The store's location should be "Seattle ID" and the store's initial funds should be \$1000. You have already loaded the manga price data into a variable called `manga_price_data`.

```
kinokuniya_seattle = MangaManager("Seattle ID", manga_price_data, 1000)
```

Q6.2 Add manga

2 Points

Write client code to add 11 copies of the manga "Blue Lock" and 7 copies of "Attack on Titan" to `kinokuniya_seattle`. This code would appear after the code you have written for the problem above.

```
kinokuniya_seattle.add_manga("Blue Lock", 11)  
kinokuniya_seattle.add_manga("Attack on Titan", 7)
```

Q6.3 What is printed?

1 Point

What will be printed if the following lines of code are executed immediately after your code from the prior two problems have been run.

```
kinokuniya_seattle.sell_manga("Blue Lock", 3)  
kinokuniya_seattle.sell_manga("Love Is War", 1)  
kinokuniya_seattle.sell_manga("Blue Lock", 10)
```

Your Answer:

```
Sold 3 copies of Blue Lock  
Not enough copies of Love Is War in stock!  
Not enough copies of Blue Lock in stock!
```

Q6.4 get_total_value

4 Points

Add a method `get_total_value` to the `MangaManager` class that will calculate and return the total value in dollars of the current `manga_stock` based on the prices in the `manga_prices` field. You may assume all manga in `manga_stock` are part of `manga_prices`. Please write the entire function including the definition and the body. You do not need to write a docstring.

Use 4 spaces to indicate an indentation if typing directly into the textbox, or copy and paste from VSCode. Regardless, please be sure that your indentation is as you would like it.

```
def get_total_value(self):
    """Returns the total value of the current manga_stock"""
    total_value = 0
    for item, count in self.manga_stock.items():
        total_value += self.manga_prices[item] * count
    return total_value
```

Q7 Interpreting Exceptions

8 Points

Note: For this question you should assume that the type of a variable does not change throughout the program.

You run a python program that produces the following Traceback:

```
Traceback (most recent call last):
  File "mystery.py", line 18, in <module>
    main()
  File "mystery.py", line 15, in main
    foo(record["x"])
  File "mystery.py", line 3, in foo
    baz(bank_account + bank_account)
  File "mystery.py", line 10, in baz
    bar(bank_account, pair[0], pair)
  File "mystery.py", line 6, in bar
    bank_account[name] = bank_account[name] + len(pair)
TypeError: list indices must be integers or slices, not str
```


Q7.1 The cause, in your own words

2 Points

Based on the information you have, in your own words, explain what is causing the error that is reported.

A TypeError is caused by the expression `bank_account[name]`, on the right hand side of the `=`, on line 6 in the function bar in the file `mystery.py`. In this expression, we are trying to access `bank_account` (a list) with a name (a string), and are being told that string is not the right type to use when indexing a list.

Q7.2 Which expression?

2 Points

Based on the information you have, what do you think is triggering this error (select all that apply)?

the expression `record["x"]`

the expression `bank_account + bank_account`

the expression `pair[0]`

the expression `bank_account[name]` on the right hand side of the assignment statement

the expression `len(pair)`

the expression `bank_account[name]` on the left hand side of the assignment statement

Q7.3 What types?

4 Points

For this question you should assume that the type of a variable does not change throughout the program.

What do you think the type of `record` might be (give all types it could be based on the information you have)?

dictionary

What do you think the type of `name` might be (give all types it could be based on the information you have)?

string

What do you think the type of `bank_account` might be (give all types it could be based on the information you have)?

list

What do you think the type of `pair` might be (give all types it could be based on the information you have)?

list, tuple, dictionary, string

Q8 Testing

6 Points

Given this docstring for the function `count_characters` (below), write three tests that each test a different thing. Each test should be written as an assert statement and should test something significantly different than your other tests.

For example, if I was testing the `abs` function, three possible tests might be:

```
assert abs(-1) == 1
assert abs(7) == 7
assert abs(0) == 0
```

```
def count_characters(input):
    '''Takes in a string as input and returns a dictionary containing
    the counts of every distinct character. Return an empty
    dictionary if the argument is an empty string.'''
```

Arguments: input: a string

...

Test #1:

```
assert count_characters("") == {}
```

What test #1 is testing:

An empty string case, to ensure the function return value matches what the documentation says to return for an empty string.

Test #2:

```
assert count_characters("a b c!") == {"a" : 1, " " : 3, "b" : 1, "c" : 1, "!": 1}
```

What test #2 is testing:

A case checking that whitespace and punctuation, both of which are considered characters, are included in the return value.

Test #3:

```
assert count_characters("aBbcCc") == {"a" : 1, "B" : 1, "b" : 1, "c": 2, "C" : 1}
```

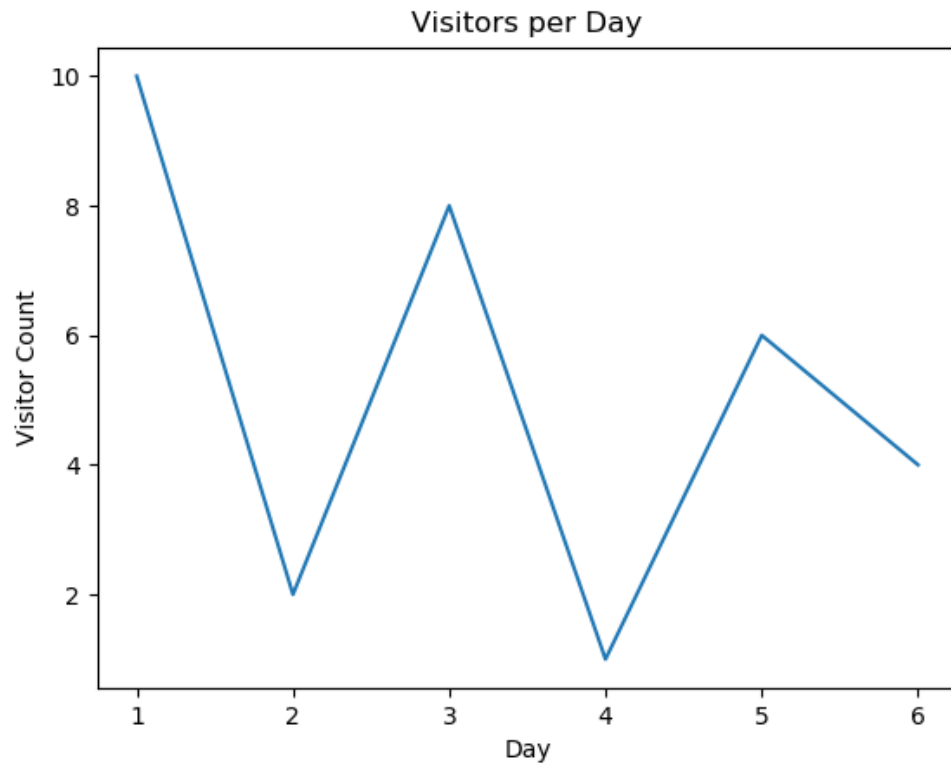
What test #3 is testing:

Checks that the upper and lowercase version of letters are considered distinct characters.

Q9 Visualization

6 Points

Given the following plot:



Write code to create and **display** this EXACT plot and **save it into the file "visitors.png"**. Note: this plot was made using only 6 integer data points. You should import whatever you need.

You do not need to write a function. Use 4 spaces to indicate an indentation if typing directly into the textbox, or copy and paste from VSCode. Regardless, please be sure that your indentation is as you would like it.

```
import matplotlib.pyplot as plt
x_vals = [1, 2, 3, 4, 5, 6] # Must have x-coord list
y_vals = [10, 2, 8, 1, 6, 4]
plt.plot(x_vals, y_vals)
plt.ylabel("Visitor Count")
plt.xlabel("Day")
plt.title("Visitors per Day")
plt.savefig("visitors.png")
plt.show()
```

Q10 Comprehensions

5 Points

Q10.1 comp 1

2 Points

Write a list comprehension that takes a list of integers `int_lst` and creates a list containing only the values from the list that are a multiple of 10, preserving the order elements appeared in the original list. **Your solution should be one line of code.**

For example, if `int_lst = [4, 20, 17, 0, 90, 34, -30, 200]`

Then your list comprehension should create the list: `[20, 0, 90, -30, 200]`

`multiples_of_ten =`

```
[x for x in int_lst if x % 10 == 0]
```

Q10.2 comp 2

3 Points

Write a dict comprehension that takes `cost_dict`, a dictionary mapping items (strings) to a price in dollars (integers), and creates a new dictionary containing only the keys that start with the letter 'z', with their price increased by 10 dollars. **Your solution should be one line of code.**

For example, if

```
cost_dict = {"zipper": 2, "boat": 5, "tiger": 17, "zebra": 24}
```

Then your dict comprehension should create the dictionary containing these keys and values: `{'zipper': 12, 'zebra': 34}`

`z_dict =`

```
{k: v + 10 for k, v in cost_dict.items() if k[0] == 'z'}
```