Basic Questions

Tags	Data Structures and Algos
Created	@October 9, 2022
Status	Data Structures and Algos
Tags 2	Work

LIST Questions

1. **Given a list of numbers, write a Python program to find the sum of all the elements in the list.?**

```
Input: arr = [2,4,5,10], i = 1, j = 3
Output: 19

Input: arr = [4,10,5,3,3], i = 3, j = 3
Output: 3
```

2. **Given an array of integers arr[] of size N and an integer, the task is to rotate the array elements to the left by d positions.?**

```
Input:
arr[] = {1, 2, 3, 4, 5, 6, 7}, d = 2
Output: 3 4 5 6 7 1 2

Input: arr[] = {3, 4, 5, 6, 7, 1, 2}, d=2
Output: 5 6 7 1 2 3 4
```

3. **Second most repeated word in a sequence in Python?**

```
Given a sequence of strings, the task is to find out the second most
repeated (or frequent) string in the given sequence.

Input : ["aaa", "bbb", "ccc", "bbb",
         "aaa", "aaa"]
Output : bbb
```

4. **Difference between two lists ?**

```
Input:
list1 = [10, 15, 20, 25, 30, 35, 40]
list2 = [25, 40, 35]

Output:
[10, 20, 30, 15]
```

5. **Print all positive numbers from given list using for loop Iterate each element in the list using for loop and check if number is greater than or equal to 0. If the condition satisfies, then only print the number.?**

```
# Python program to print positive Numbers in a List
Input: list1 = [12, -7, 5, 64, -14]
Output: 12, 5, 64

Input: list2 = [12, 14, -95, 3]
Output: [12, 14, 3]
```

6. Write a Python program to flatten a given nested list structure.?

```
Original list: [0, 10, [20, 30], 40, 50, [60, 70, 80], [90, 100, 110, 120]]
Flatten list:
[0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120]
```

7. **Given an array and a value, find if there is a triplet in array whose sum is equal to the given value. If there is such a triplet present in array, then print the triplet and return true. Else return false.?**

```
Input: array = [12, 3, 4, 1, 6, 9], sum = 24;
Output: 12, 3, 9
Explanation: There is a triplet (12, 3 and 9) present
in the array whose sum is 24.

Input: array = [1, 2, 3, 4, 5], sum = 9
Output: 5, 3, 1
Explanation: There is a triplet (5, 3 and 1) present
in the array whose sum is 9.
```

Strings

1. **Missing characters to make a string Pangram.?**

```
Pangram is a sentence containing every letter in the English alphabet.
Given a string, find all characters that are missing from the string, i.e.,
the characters that can make the string a Pangram.
We need to print output in alphabetic order.

Input : welcome to geeksforgeeks
Output : abdhijnpquvxyz

Input : The quick brown fox jumps
Output : adglvyz
```

2. **Find total number of non-empty substrings of a string with N characters.?**

```
Input : str = “abc”
Output : 6
Every substring of the given string : “a”, “b”, “c”, “ab”, “bc”, “abc”

Input : str = “abcd”
Output : 10
Every substring of the given string : “a”, “b”, “c”, “d”, “ab”,
                                      “bc”, “cd”, “abc”, “bcd” and “abcd”
```

3. **Given a string containing lowercase and uppercase letters. Sort it in such a manner that the uppercase and lowercase letters come in an alternate manner but in a sorted way.?**

```
Input : bAwutndekWEdkd
Output :AbEdWddekkntuw
Explanation:
Here we can see that letter ‘A’, ’E’, ’W’ are sorted
as well as letters “b, d, d, d, e, k, k, n, t, u, w” are sorted
but both appears alternately in the string as far as possible.

Input :abbfDDhGFBvdFDGBNDasZVDFjkb
Output :BaBaDbDbDbDdDfFhFjFkGsGvNVZ
```

4. Write a Python program that accepts a comma separated sequence of words as input and prints the unique words in sorted form (alphanumerically). ?

```
Sample Words : red, white, black, red, green, black
Expected Result : black, green, red, white,red
```

5. Write a Python program to count the number of characters (character frequency) in a string. ?

```
Sample String : google.com'
Expected Result : {'g': 2, 'o': 3, 'l': 1, 'e': 1, '.': 1, 'c': 1, 'm': 1}
```

6. **find the frequency of minimum occurring character in a python string ?**

```
The original string is : iNeuronNet.com
The minimum of all characters in GeeksforGeeks is : i
```

8 Write a program extract all the string characters which have odd number of occurrences.

```
The original string is : geekforgeeks is best for geeks
The Odd Frequency Characters are : ['k', 'i', 't', 'g', 'e', 'b']
```

Dictionary

1. **Given an input string and a pattern, check if characters in the input string follows the same order as determined by characters present in the pattern. Assume there won’t be any duplicate characters in the pattern.?**

```
Input:
string = "engineers rock"
pattern = "er";
Output: true
Explanation:
All 'e' in the input string are before all 'r'.


Input:
string = "engineers rock"
pattern = "gsr";
Output: false
Explanation:
There are one 'r' before 's' in the input string.
```

2. **Given a list and dictionary, map each element of list with each item of dictionary, forming nested dictionary as value.?**

```
Input : test_dict = {‘Gfg’ : 4, ‘best’ : 9}, test_list = [8, 2]
Output : {8: {‘Gfg’: 4}, 2: {‘best’: 9}}
Explanation : Index-wise key-value pairing from list [8] to dict {‘Gfg’ : 4} and so on.
```

3. **Sort Dictionary key and values List ?**

```
Input : test_dict = {‘c’: [3], ‘b’: [12, 10], ‘a’: [19, 4]}
Output : {‘a’: [4, 19], ‘b’: [10, 12], ‘c’: [3]}

Input : test_dict = {‘c’: [10, 34, 3]}
Output : {‘c’: [3, 10, 34]}
```

4. **Remove all duplicates words from a given sentence ?**

```
Input : Python is great and Java is also great
Output : is also Java Python and great
```

5. **Inversion in nested dictionary?**

```
Input : test_dict = {“a” : {“b” : {}},
“d” : {“e” : {}},
“f” : {“g” : {}}
Output : {‘b’: {‘a’: {}}, ‘e’: {‘d’: {}}, ‘g’: {‘f’: {}}
Explanation : Nested dictionaries inverted as outer dictionary keys and viz-a-vis.
```

6. **Given an array of n string containing lowercase letters. Find the size of largest subset of string which are anagram of each others. An anagram of a string is another string that contains same characters, only the order of characters can be different. For example, “abcd” and “dabc” are anagram of each other.?**

```
Input:
ant magenta magnate tan gnamate
Output: 3
Explanation
Anagram strings(1) - ant, tan
Anagram strings(2) - magenta, magnate,
                     gnamate
Thus, only second subset have largest
size i.e., 3

Input:
cars bikes arcs steer
Output: 2
```

Sets

1. Given two lists a, b. Check if two lists have at least one element common in them.

```
Input : a = [1, 2, 3, 4, 5]
        b = [5, 6, 7, 8, 9]
Output : True

Input : a=[1, 2, 3, 4, 5]
        b=[6, 7, 8, 9]
Output : False
```

2. **Return a new set of identical items from two sets**

```
set1 = {10, 20, 30, 40, 50}
set2 = {30, 40, 50, 60, 70}
Expected output:

{40, 50, 30}
```

3. **Maximum and Minimum in a Set without use of inbuild max/min functions?**

```
Input : set = ([8, 16, 24, 1, 25, 3, 10, 65, 55])
Output : max is 65

Input : set = ([4, 12, 10, 9, 4, 13])
Output : min is 4
```

4. Write a Python program to check if a set is a subset of another set

```
Input :-
x:  {'mango', 'apple'}
y:  {'mango', 'orange'}
z:  {'mango'}

output :-
If x is subset of y
False
False
If y is subset of x
False
False

If y is subset of z
False
False
If z is subset of y
```

```
True
True
```

5. Write a Python program to remove the intersection of a 2nd set from the 1st set

```
input:-
Original sets:
{1, 2, 3, 4, 5}
{4, 5, 6, 7, 8}
Output:-
sn1:  {1, 2, 3}
sn2:  {4, 5, 6, 7, 8}
```

6. **What is the result of passing a dictionary to a set constructor?**

Tuple

1. **Remove Tuples of Length K ?**

```
Input : test_list = [(4, 5), (4, ), (8, 6, 7), (1, ), (3, 4, 6, 7)], K = 2
Output : [(4, ), (8, 6, 7), (1, ), (3, 4, 6, 7)]
Explanation : (4, 5) of len = 2 is removed.
```

2. **Removing duplicates from tuple ?**

```
The original tuple is : (1, 3, 5, 2, 3, 5, 1, 1, 3)
The tuple after removing duplicates : (1, 3, 5, 2)
```

3. **Flatten tuple of List to tuple ?**

```
Input : test_tuple = ([5], [6], [3], [8]) Output : (5, 6, 3, 8)
Input : test_tuple = ([5, 7, 8]) Output : (5, 7, 8)
```

4. **Remove nested records from tuple?**

```
The original tuple : (1, 5, 7, (4, 6), 10)
Elements after removal of nested records : (1, 5, 7, 10)
```

5. **Convert Binary tuple to Integer?**

```
The original tuple is : (1, 1, 0, 1, 0, 0, 1)
Decimal number is : 105
```

6. **Sort Tuples by Total digits?**

```
Input : test_list = [(3, 4, 6, 723), (1, 2), (134, 234, 34)]
Output : [(1, 2), (3, 4, 6, 723), (134, 234, 34)]
Explanation : 2 < 6 < 8, sorted by increasing total digits.
```

MAP & Lambda Function

1. Write a Python program to add three given lists using Python map and lambda.?

```
Original list:
[1, 2, 3]
[4, 5, 6]
[7, 8, 9]

New list after adding above three lists:
[12, 15, 18]
```

2. Write a Python program to convert a given list of tuples to a list of strings using map function?

```
Original list of tuples:
[('red', 'pink'), ('white', 'black'), ('orange', 'green')]

Convert the said list of tuples to a list of strings:
['red pink', 'white black', 'orange green']
```

3. Write a Python program to add two given lists and find the difference between lists. Use map() function?

```
Original lists:
[6, 5, 3, 9]
[0, 1, 7, 7]

Result:
[(6, 6), (6, 4), (10, -4), (16, 2)]
```

4. Write a Python program to create Fibonacci series upto n using Lambda?

```
Fibonacci series upto 2:
[0, 1]
Fibonacci series upto 5:
[0, 1, 1, 2, 3]
```

5. Write a Python program to find intersection of two given arrays using Lambda ?

```
[1, 2, 3, 5, 7, 8, 9, 10]
[1, 2, 4, 8, 9]
Intersection of the said arrays: [1, 2, 8, 9]
```

6. Write a Python program to find palindromes in a given list of strings using Lambda ?

```
Orginal list of strings:
['php', 'w3r', 'Python', 'abcd', 'Java', 'aaa']
List of palindromes:
['php', 'aaa']
```

7. Write a Python program to find the list with maximum and minimum length using lambda ?

```
Original list:
[[0], [1, 3], [5, 7], [9, 11], [13, 15, 17]]
List with maximum length of lists:
(3, [13, 15, 17])
List with minimum length of lists:
(1, [0])
```

8. Write a Python program to triple all numbers of a given list of integers.?

```
Original list:  (1, 2, 3, 4, 5, 6, 7)

Triple of said list numbers:
[3, 6, 9, 12, 15, 18, 21]
```

List Comprehension

1. Use a nested list comprehension to find all of the numbers from 1–1000 that are divisible by any single digit besides 1 (2–9)?

2. Use list comprehension to construct a new list but add 6 to each item?
3. Suppose we want to create an output dictionary which contains only the odd numbers that are present in the input list as keys and their cubes as values. Let's see how to do this using for loops and dictionary comprehension.?

```
input = [1, 2, 3, 4, 5, 6, 7]
outpur :- {1: 1, 3: 27, 5: 125, 7: 343}
```

4. Given two lists containing the names of states and their corresponding capitals, construct a dictionary which maps the states with their respective capitals. Let's see how to do this using for loops and dictionary comprehension.?

```
state = ['Gujarat', 'Maharashtra', 'Rajasthan']
capital = ['Gandhinagar', 'Mumbai', 'Jaipur']
output:- {'Gujarat': 'Gandhinagar','Maharashtra': 'Mumbai', 'Rajasthan': 'Jaipur'}
```

5. **Transpose of Matrix using Comprehension?**

```
Input :- [[1, 2, 3, 4], [4, 5, 6, 8]]
Output :- [[1, 4], [2, 5], [3, 6], [4, 8]]
```

6. We have a string '2459a09b' and we want to extract all integer literals, and use int() to cast them into integers.?

```
Input- 2459a09b'
output :- [2, 4, 5, 9, 9]
```

7. **Finding the elements in a list in which elements are ended with the letter 'b' and the length of that element is greater than 2?**

```
input :- names = ['Ch','Dh','Eh','cb','Tb','Td','Chb','Tdb']
output :- ['Chb', 'Tdb']
```

8. **Reverse each String in a Tuple using list comprehension ?**

```
Input :- 'Hello', 'Analytics', 'Vidhya'
output :- ['olleH', 'scitylanA', 'ayhdiV']
```

Object Oriented Programming

1. **Define a property that must have the same value for every class instance (object)?**

```
Output :-
Color: White, Vehicle name: School Volvo, Speed: 180, Mileage: 12
Color: White, Vehicle name: Audi Q5, Speed: 240, Mileage: 18
```

2. Create a class with property decorator with setter and getter functions.?
3. Create a class with Multi-level Inheritance ?
4. Create a class a overwrite a class Destructor ?
5. write a **decorator for a class?**
6. Implement a stack ?
7. Implement a Queue ?
8. Implement a Linked List ?
9. Explain What happens you create a object of a class (Inside python)?
10. Create a class whose object should be called like a function.?

```
class Product:
    def __init__(self):
        print("Instance Created")

# Instance created
ans = Product()
ans(10, 20) -> should return product
```

Exception Handling

1. Write Custom Exception to handle Zero division Error ?
2. Catching specific exception in Python.?