Questions on List PART I

Shubham Verma

List Theory https://lnkd.in/ezAmJb4U

Linkedin https://www.linkedin.com/in/shubham-verma-3968a5119

GitHub https://lnkd.in/gky-wyFJ

Credits W3School for questions

1. Write a Python program to sum all the items in a list.

In [1]:
```
l = [i for i in range(10)]
def sum(input_list):
    sum=0
    for i in input_list:
        sum += i
    return sum
sum(l)
```

Out[1]:
```
45
```

2. Write a Python program to multiply all the items in a list.

In [3]:
```
l = [i for i in range(1,10)]
def multiply(input_list):
    product=1
    for i in input_list:
        product *= i
    return product
multiply(l)
```

Out[3]:
```
362880
```

3. Write a Python program to get the largest number from a list.

In [7]:
```
l = [8,7213,12414,987,123456,343]
def largest_number(input_list):
    return sorted(input_list)[len(input_list)-1]
largest_number(l)
```

Out[7]:
```
123456
```

4. Write a Python program to get the smallest number from a list.

In [9]:
```
l = [8,7213,12414,987,123456,343]
def smallest_number(input_list):

```

```
    return sorted(input_list)[0]
smallest_number(l)
```

Out[9]: 8

5. Write a Python program to count the number of strings where the string length is 2 or more and the first and last character are same from a given list of strings.

In [10]:
```
l_string = ['abc', 'xyz', 'aba', '1221']
def custom_string_count(input_list):
    count = 0
    for string in input_list:
        if len(string)>=2 and string[0] == string[-1]:
            count += 1
    return count
custom_string_count(l_string)
```

Out[10]: 2

6. Write a Python program to get a list, sorted in increasing order by the last element in each tuple from a given list of non-empty tuples.

In [12]:
```
l_sample = [(2, 5), (1, 2), (4, 4), (2, 3), (2, 1)]
def sort_custom(input_list):
    return sorted(input_list, key=lambda x : x[1])

sort_custom(l_sample)
```

Out[12]: [(2, 1), (1, 2), (2, 3), (4, 4), (2, 5)]

7. Write a Python program to remove duplicates from a list.

In [13]:
```
l_sample=[1,1,2,2,3,4,4,55,6,6,7,87,8,9,99,0]
def duplicate_remover(input_list):
    return list(set(input_list))

duplicate_remover(l_sample)
```

Out[13]: [0, 1, 2, 3, 4, 99, 6, 7, 8, 9, 87, 55]

8. Write a Python program to check a list is empty or not.

In [14]:
```
l_empty = []
l_sample = [1,"shubham"]
def list_checker_len(input_list):
    if len(input_list) == 0:
        return "Input list is empty"
    else:
        return "Input list is not empty"

list_checker_len(l_sample)
```

Out[14]: 'Input list is not empty'

In [15]:
```
list_checker_len(l_empty)
```

Out[15]:
```
'Input list is empty'
```

9. Write a Python program to find the list of words that are longer than n from a given list of words.

In [16]:
```
def custom_word_length(input_list, length_of_word):
    l = []
    for word in input_list:
        if len(word) > length_of_word:
            l.append(word)
    return l

list_sample = ['i', 'am', 'a', 'data', 'science', 'enthusiast']
custom_word_length(list_sample,2)
```

Out[16]:
```
['data', 'science', 'enthusiast']
```

10. Write a Python function that takes two lists and returns True if they have at least one common member.

In [19]:
```
l1 = [1,2,3,4,55]
l2 = [2,3,5,6,7,8]
l3 = ["shub","simran"]

def common_checker_lists(list1, list2):
    count=0
    for member in list1:
        if member in list2:
            count+=1
    if count>0:
        return "True"
    else:
        return "False"

common_checker_lists(l1,l2)
```

Out[19]:
```
'True'
```

In [20]:
```
common_checker_lists(l2,l3)
```

Out[20]:
```
'False'
```

11. Write a Python program to print a specified list after removing the 0th, 4th and 5th elements.

In [31]:
```
list_colours = ['Red', 'Green', 'White', 'Black', 'Pink', 'Yellow']

def element_remover_list(input_list, *index):
    l=[]
    for i in range(len(input_list)):
        if i not in index:
            l.append(input_list[i])
    return l

element_remover_list(list_colours, 0,4,5)
```

Out[31]:
```
['Green', 'White', 'Black']
```

12. Write a Python program to print the numbers of a specified list after removing even numbers from it.

In [4]:
```
list_int= [i for i in range(15)]

def even_remover(input_list):
    l=[]
    for i in input_list:
        if i%2 !=0 and i != 0:
            l.append(i)
    return l

even_remover(list_int)
```

Out[4]:
```
[1, 3, 5, 7, 9, 11, 13]
```

13. Write a Python program to shuffle and print a specified list.

In [30]:
```
from random import shuffle
l_int = [i for i in range(10)]

def list_shuffle(input_list):
    shuffle(input_list)
    return input_list

list_shuffle(l_int)
```

Out[30]:
```
[2, 3, 5, 7, 0, 9, 1, 8, 4, 6]
```

14. Write a Python program to generate and print a list of first and last 5 elements square in range of numbers between 1 and 30 (both included).

In [32]:
```
lint = [i for i in range(1,31)]

def custom_square(input_list):
    l=[]
    for i in input_list[:5]+input_list[-5:]:
        l.append(i**2)
    return l

custom_square(lint)
```

Out[32]:
```
[1, 4, 9, 16, 25, 676, 729, 784, 841, 900]
```

15. Write a Python program to generate all permutations of a list in Python.

In [34]:
```
import itertools

l=['Red', 'Yellow','Brown']

def list_permutations(input_list):
```

```
    return list(itertools.permutations(input_list))

list_permutations(l)
```

Out[34]:
```
[('Red', 'Yellow', 'Brown'),
 ('Red', 'Brown', 'Yellow'),
 ('Yellow', 'Red', 'Brown'),
 ('Yellow', 'Brown', 'Red'),
 ('Brown', 'Red', 'Yellow'),
 ('Brown', 'Yellow', 'Red')]
```

16.Write a Python program to get the difference between the two lists.

In [8]:
```
l1 = [1,2,3,4,55]
l2 = [2,3,5,5,6,7,8]
l3 = ["shub","simran"]
l4 = ['simran',"Shashank",'shobit']

def difference_bw_lists(list1, list2):
    l=[]
    for member in list1:
        if member in list2:
            pass
        else:
            l.append(member)
    for member in list2:
        if member in list1:
            pass
        else:
            l.append(member)
    return list(l)

difference_bw_lists(l1,l2)
```

Out[8]:
```
[1, 4, 55, 5, 5, 6, 7, 8]
```

In [7]:
```
difference_bw_lists(l3,l4)
```

Out[7]:
```
['shub', 'Shashank', 'shobit']
```

17.Write a Python program to get the difference between the two lists(without duplicates).

In [10]:
```
l1 = [1,2,3,4,55]
l2 = [2,3,5,5,6,7,8]


def difference_bw_lists1(list1, list2):
    l=[]
    for member in list1:
        if member in list2:
            pass
        else:
            l.append(member)
    for member in list2:
        if member in list1:
            pass
        else:
            l.append(member)
    return list(set(l))
```

```
difference_bw_lists1(l1,l2)
```

Out[10]:
```
[1, 4, 5, 6, 7, 8, 55]
```

18. Write a Python program access the index of a list.

In [13]:
```
l = ['Red', 'Green', 'White', 'Black', 'Pink', 'Yellow']

def index_extractor(input_list):
    return [i for i in range(len(input_list))]

index_extractor(l)
```

Out[13]:
```
[0, 1, 2, 3, 4, 5]
```

19. Write a Python program to convert a list of characters into a string.

In [14]:
```
l = ["i", "N", "e", "u", "r","o","n"]

def list_char_to_string(input_list):
    return "".join(input_list)

list_char_to_string(l)
```

Out[14]:
```
'iNeuron'
```

20. Write a Python program to find the index of an item in a specified list.

In [28]:
```
l = ['Red', 'Green', 'White', 'Black', 'Pink', 'Yellow']

def specific_index_list(input_list, input_item):
    count=-1
    for i in input_list:
        count+=1
        if i == input_item:
            break
    return "index of {} is {}".format(input_item,count)

specific_index_list(l,'Yellow')
```

Out[28]:
```
'index of Yellow is 5'
```

In [29]:
```
specific_index_list(l,'Red')
```

Out[29]:
```
'index of Red is 0'
```

21. Write a Python program to flatten a shallow list.

In [30]:
```
list_shallow = [[1,2,3],[4,5,6],[7,8,9]]

def list_flattner(input_list):
    l=[]
    for i in input_list:
        l += i
```

```
    return l

list_flattner(list_shallow)
```

Out[30]: [1, 2, 3, 4, 5, 6, 7, 8, 9]

22. Write a Python program to append a list to the second list.

In [31]:
```
l1 = [1,2,3,45,4]
l2 = [44,55,66,77,88]

def list_appender(in_list1, in_list2):
    return in_list1+in_list2

list_appender(l1,l2)
```

Out[31]: [1, 2, 3, 45, 4, 44, 55, 66, 77, 88]

23. Write a Python program to select an item randomly from a list.

In [38]:
```
import random

list1 = [1,2,3,4,5,"sameer", "samira"]

def item_random_list(input_list):
    return random.choice(input_list)

item_random_list(list1)
```

Out[38]: 'sameer'

24.Write a Python program to find the second smallest number in a list.

In [40]:
```
l = [33,45,67,89,1,2,3,4,55,66,789]

def second_smallest(input_list):
    return sorted(input_list)[1]

second_smallest(l)
```

Out[40]: 2

25. Write a Python program to find the second largest number in a list.

In [42]:
```
l = [33,45,67,89,1,2,3,4,55,686,789]

def second_largest(input_list):
    return sorted(input_list, reverse=True)[1]

second_largest(l)
```

Out[42]: 686

26. Write a Python program to get unique values from a list.

In [43]:
```
l = [1,1,2,2,3,3,4,4,5,5,5,6,6,6,88]

def list_unique(input_list):
    return list(set(input_list))

list_unique(l)
```

Out[43]:
```
[1, 2, 3, 4, 5, 6, 88]
```

27. Write a Python program to get the frequency of the elements in a list.

In [46]:
```
list1 = [1,1,2,2,3,3,4,4,5,5,5,6,6,6,88]

def element_counter(input_list):
    l = []
    for i in set(input_list):
        l.append((i, input_list.count(i)))
    return l

element_counter(list1)
```

Out[46]:
```
[(1, 2), (2, 2), (3, 2), (4, 2), (5, 3), (6, 3), (88, 1)]
```

28. Write a Python program to count the number of elements in a list within a specified range.

In [49]:
```
l = [1,2,3,4,5,6,7,8,9,10]

def element_counter_range(input_list, start, end):
    count = 0
    for element in input_list:
        if element > start and element < end:
            count +=1
    return count

element_counter_range(l,1,10)
```

Out[49]:
```
8
```

29. Write a Python program to check whether a list contains a sublist.

In [51]:
```
l= [[1,2,3],4,6,'shubham']
l2 = [1,2,3,4,5,6]

def sublist_checker(input_list):
    for i in input_list:
        if type(i) == list:
            return "Sublist present inside the given list"
        else:
            return "Sublist is not present inside the given list"

sublist_checker(l)
```

Out[51]:
```
'Sublist present inside the given list'
```

In [52]:
```
sublist_checker(l2)
```

Out[52]:
```
'Sublist is not present inside the given list'
```

30. Write a Python program to generate all sublists of a list.

In [53]:
```
list1 = ["shubham", "sameer",1,2]

from itertools import combinations
def sublist_finder(input_list):
    l = []
    for i in range(0, len(input_list)+1):
        temp = [list(x) for x in combinations(input_list,i)]
        if len(temp)>0:
            l.extend(temp)
    return l

sublist_finder(list1)
```

Out[53]:
```
[[],
 ['shubham'],
 ['sameer'],
 [1],
 [2],
 ['shubham', 'sameer'],
 ['shubham', 1],
 ['shubham', 2],
 ['sameer', 1],
 ['sameer', 2],
 [1, 2],
 ['shubham', 'sameer', 1],
 ['shubham', 'sameer', 2],
 ['shubham', 1, 2],
 ['sameer', 1, 2],
 ['shubham', 'sameer', 1, 2]]
```

31. Write a Python program to convert list to dictionary.

In [57]:
```
name = ["Shubham", "Krish", "Sudhanshu"]
surname = ["Verma","Naik","Kumar"]

def list_to_dict(in_l1,in_l2):
    d = {}
    for key in in_l1:
        for value in in_l2:
            d[key] = value
    return d

list_to_dict(name, surname)
```

Out[57]:
```
{'Shubham': 'Kumar', 'Krish': 'Kumar', 'Sudhanshu': 'Kumar'}
```

32. Write a Python program to create a list by concatenating a given list which range goes from 1 to n.

In [60]:
```
list1 = ['india','russia']
```

```
def list_custom(input_list,n):
    return ["{} {}".format(a,b) for a in input_list for b in range(1,n+1) ]

list_custom(list1,3)
```

Out[60]:
```
['india 1', 'india 2', 'india 3', 'russia 1', 'russia 2', 'russia 3']
```

33. Write a Python program to sort a list of nested dictionaries.

In [82]:
```
d = [{'Name':{'Surname': 3}}, {'Name':{'Surname': 2}}, {'Name':{'Surname': 99}}]

def list_nested_dict_sort(in_dict):
    in_dict.sort(key=lambda x: x['Name']['Surname'])
    return in_dict

list_nested_dict_sort(d)
```

Out[82]:
```
[{'Name': {'Surname': 2}}, {'Name': {'Surname': 3}}, {'Name': {'Surname': 99}}]
```

34. Write a Python program to find common items from two lists.

In [83]:
```
name = ["shubham", "sameer", "pratiksha"]
name1 = ["shubham","prathiba","sameer"]

def common_list_item(in_l1, in_l2):
    l = []
    for i in in_l1:
        if i in in_l2:
            l.append(i)
    return l

common_list_item(name, name1)
```

Out[83]:
```
['shubham', 'sameer']
```

35. Write a Python program to split a list in N parts.

In [90]:
```
l = [1,2,3,4,5,6,7,8,9, 10]

def custom_split_element(in_list, element_split):
    return [in_list[i::element_split] for i in range(element_split)]

custom_split_element(l,2)
```

Out[90]:
```
[[1, 3, 5, 7, 9], [2, 4, 6, 8, 10]]
```

36. Write a Python program to convert a list of multiple integers into a single integer.

In [93]:
```
l = [11,22,33,44]

def list_int_to_int(input_list):
    string = ""
    for i in input_list:
        string+= str(i)
```

```
    return string

list_int_to_int(l)
```

Out[93]: '11223344'

37. Write a Python program to split a list based on first character of word and return character and word.

In [95]:
```
l = ["Shubham", "Sameer", "Sushant"]

def custom_split_1(in_list):
    for i in in_list:
        print(i[0], i)

custom_split_1(l)
```

```
S Shubham
S Sameer
S Sushant
```

38. Write a Python program to create dictionary multiple empty lists.

In [98]:
```
def empty_list_gen(n):
    d={}
    for i in range(1,n+1):
        d[i]= []
    return d

empty_list_gen(4)
```

Out[98]: {1: [], 2: [], 3: [], 4: []}

39. Write a Python program to find missing and additional values in two lists.

In [103…
```
l1 = ['a','b','c','d','e','f']
l2 = ['d','e','f','g','h']

def custom_return_list(in_l1, in_l2):
    return "missing values: {} and additional values: {}".format(','.join(set(in_l1

custom_return_list(l1,l2)
```

Out[103]: 'missing values: c,a,b and additional values: g,h'

40. Write a Python program to check if all items of a given list of strings is equal to a given string.

In [105…
```
l1 = [1,2,3,4,5]
l2 = [1,1,1,1,1]

def list_string_checker(input_list, element):
    return all(x == element for x in input_list)

list_string_checker(l1,1)
```

Out[105]:
```
False
```

In [106…
```
list_string_checker(l2,1)
```

Out[106]:
```
True
```

41. Write a Python program to generate groups of five consecutive numbers in a list.

In [110…
```
def consecutive_list_gen(start_number):
    return [[5*i + j for j in range(start_number,start_number +5)] for i in range(

consecutive_list_gen(5)
```

Out[110]:
```
[[5, 6, 7, 8, 9],
 [10, 11, 12, 13, 14],
 [15, 16, 17, 18, 19],
 [20, 21, 22, 23, 24],
 [25, 26, 27, 28, 29]]
```

42. Write a Python program to convert a pair of values into a sorted unique array.

In [111…
```
l_int= [(1,2),(2,3),(3,4),(4,5)]

def custom_list_sorted_unique(input_list):
    return sorted(set().union(*input_list))

custom_list_sorted_unique(l_int)
```

Out[111]:
```
[1, 2, 3, 4, 5]
```

43. Write a Python program to select the odd items of a list.

In [113…
```
l = ['Red', 'Blue', 'Green', 'Pink', 'Brown']

def odd_element_picker_list(input_list):
    return input_list[::2]

odd_element_picker_list(l)
```

Out[113]:
```
['Red', 'Green', 'Brown']
```

44. Write a Python program to insert an element before each element of a list.

In [115…
```
list1 =[1,2,3,4,5]

def element_insert(in_list, element):
    return [comb for i in in_list for comb in (element, i)]

element_insert(list1, "sameer")
```

Out[115]:
```
['sameer', 1, 'sameer', 2, 'sameer', 3, 'sameer', 4, 'sameer', 5]
```

45. Write a Python program to print a nested lists (each list on

a new line) using the print() function.

In [119…
```
names = [['Shubham'], ['Sameer'], ['Brijesh'], ['Sandesh']]

def custom_print_list(in_list):
    print('\n'.join([str(i) for i in in_list]))

custom_print_list(names)
```

```
['Shubham']
['Sameer']
['Brijesh']
['Sandesh']
```

46. Write a Python program to convert list to list of dictionaries.

In [56]:
```
name = ["Shubham", "Krish", "Sudhanshu"]
surname = ["Verma","Naik","Kumar"]

def list_to_listdict(in_l1, in_l2):
    return [{'key': key, 'value': value} for key,value in zip(in_l1, in_l2)]

list_to_listdict(name, surname)
```

Out[56]:
```
[{'key': 'Shubham', 'value': 'Verma'},
 {'key': 'Krish', 'value': 'Naik'},
 {'key': 'Sudhanshu', 'value': 'Kumar'}]
```

47. Write a Python program to compute the difference between two lists.

In [120…
```
l1 = [1,2,3,4,5]
l2 = [2,3,4,5,6]

def difference_list(in_l1, in_l2):
    l=[]
    for i in in_l1:
        if i not in in_l2:
            l.append(i)
    for i in in_l2:
        if i not in in_l1:
            l.append(i)
    return l

difference_list(l1,l2)
```

Out[120]:
```
[1, 6]
```

48. Write a Python program to concatenate elements of a list.

In [123…
```
l = [1,2,3,4,5]
l_str = ["Shubham", "Verma"]

def concatenate_list_element(in_list):
    string = ""
    for i in in_list:
        if type(i) == int:
            string += str(i)
        else:
```

```
            string += str(i)

    return string

concatenate_list_element(l_str)
```

Out[123]: 'ShubhamVerma'

In [124…
```
concatenate_list_element(l)
```

Out[124]: '12345'

49. Write a Python program to check if all items of a given list of strings is equal to a given string.

In [128…
```
l = ["Cat","Hat", "Bat", "Mat"]
l1 = ["Cat","Hat", "Bats", "Mats"]

def string_length_checker(in_list, element):
    return all(len(x) == len(element) for x in in_list)

string_length_checker(l, "car")
```

Out[128]: True

In [127…
```
string_length_checker(l1,"car")
```

Out[127]: False

50. Write a Python program to replace the last element in a list with another list.

In [130…
```
l = [1,2,3,4,5,6,7,8,9,10]
l_to_replace = ["Red", "Blue", "Green"]

def custom_replace(in_list, rep_list):
    in_list[-1:] = rep_list
    return in_list

custom_replace(l, l_to_replace)
```

Out[130]: [1, 2, 3, 4, 5, 6, 7, 8, 9, 'Red', 'Blue', 'Green']