**Assinment on python 17/12/22**

**Python Problems**

**Exercise 1: Create a list by picking an odd-index items from the first list and even index items from the second**

Given two lists, l1 and l2, write a program to create a third list l3 by picking an odd-index element from the list l1 and even index elements from the list l2.

**Given**:

l1 = [3, 6, 9, 12, 15, 18, 21]

l2 = [4, 8, 12, 16, 20, 24, 28]

**Expected Output**:

Element at odd-index positions from list one

[6, 12, 18]

Element at even-index positions from list two

[4, 12, 20, 28]

Printing Final third list

[6, 12, 18, 4, 12, 20, 28]

ans

list1 = [3, 6, 9, 12, 15, 18, 21]

list2 = [4, 8, 12, 16, 20, 24, 28]

res = **list**()

odd\_elements = list1[1::2]

**print**("Element at odd-index positions from list one")

**print**(odd\_elements)

even\_elements = list2[0::2]

**print**("Element at even-index positions from list two")

**print**(even\_elements)

**print**("Printing Final third list")

res.extend(odd\_elements)

res.extend(even\_elements)

**print**(res)

**Exercise 2: Remove and add item in a list**

Write a program to remove the item present at index 4 and add it to the 2nd position and at the end of the list.

**Given**:

list1 = [54, 44, 27, 79, 91, 41]

**Expected Output**:

List After removing element at index 4  [34, 54, 67, 89, 43, 94]

List after Adding element at index 2  [34, 54, 11, 67, 89, 43, 94]

List after Adding element at last  [34, 54, 11, 67, 89, 43, 94, 11]

Ans

sample\_list = [34, 54, 67, 89, 11, 43, 94]

**print**("Original list ", sample\_list)

element = sample\_list.pop(4)

**print**("List After removing element at index 4 ", sample\_list)

sample\_list.insert(2, element)

**print**("List after Adding element at index 2 ", sample\_list)

sample\_list.append(element)

**print**("List after Adding element at last ", sample\_list)

**Exercise 3: Slice list into 3 equal chunks and reverse each chunk**

**Given**:

sample\_list = [11, 45, 8, 23, 14, 12, 78, 45, 89]

**Expected Outcome**:

Chunk  1 [11, 45, 8]

After reversing it  [8, 45, 11]

Chunk  2 [23, 14, 12]

After reversing it  [12, 14, 23]

Chunk  3 [78, 45, 89]

After reversing it  [89, 45, 78]

Ans

sample\_list = [11, 45, 8, 23, 14, 12, 78, 45, 89]

**print**("Original list ", sample\_list)

length = **len**(sample\_list)

chunk\_size = **int**(length / 3)

start = 0

end = chunk\_size

# run loop 3 times

**for** i **in** **range**(3):

# get indexes

indexes = **slice**(start, end)

# get chunk

list\_chunk = sample\_list[indexes]

**print**("Chunk ", i, list\_chunk)

# reverse chunk

**print**("After reversing it ", **list**(**reversed**(list\_chunk)))

start = end

end += chunk\_size

**Exercise 4: Count the occurrence of each element from a list**

Write a program to iterate a given list and count the occurrence of each element and create a dictionary to show the count of each element.

**Given**:

sample\_list = [11, 45, 8, 11, 23, 45, 23, 45, 89]

**Expected Output**:

Printing count of each item   {11: 2, 45: 3, 8: 1, 23: 2, 89: 1}

sample\_list = [11, 45, 8, 11, 23, 45, 23, 45, 89]

**print**("Original list ", sample\_list)

count\_dict = **dict**()

**for** item **in** sample\_list:

**if** item **in** count\_dict:

count\_dict[item] += 1

**else**:

count\_dict[item] = 1

**print**("Printing count of each item ", count\_dict)

**Exercise 5: Create a Python set such that it shows the element from both lists in a pair**

**Given**:

first\_list = [2, 3, 4, 5, 6, 7, 8]

second\_list = [4, 9, 16, 25, 36, 49, 64]

**Expected Output**:

Result is  {(6, 36), (8, 64), (4, 16), (5, 25), (3, 9), (7, 49), (2, 4)}

 first\_list = [2, 3, 4, 5, 6, 7, 8]

**print**("First List ", first\_list)

second\_list = [4, 9, 16, 25, 36, 49, 64]

**print**("Second List ", second\_list)

result = **zip**(first\_list, second\_list)

result\_set = **set**(result)

**print**(result\_set)

**Exercise 6: Find the intersection (common) of two sets and remove those elements from the first set**

**Given**:

first\_set = {23, 42, 65, 57, 78, 83, 29}

second\_set = {57, 83, 29, 67, 73, 43, 48}

**Expected Output**:

Intersection is  {57, 83, 29}

First Set after removing common element  {65, 42, 78, 23}

**Exercise 7: Checks if one set is a subset or superset of another set. If found, delete all elements from that set**

**Given**:

first\_set = {27, 43, 34}

second\_set = {34, 93, 22, 27, 43, 53, 48}

**Expected Output**:

First set is subset of second set - True

Second set is subset of First set -  False

First set is Super set of second set -  False

Second set is Super set of First set -  True

First Set  set()

Second Set  {67, 73, 43, 48, 83, 57, 29}

Ans

first\_set = {23, 42, 65, 57, 78, 83, 29}

second\_set = {57, 83, 29, 67, 73, 43, 48}

**print**("First Set ", first\_set)

**print**("Second Set ", second\_set)

intersection = first\_set.intersection(second\_set)

**print**("Intersection is ", intersection)

**for** item **in** intersection:

first\_set.remove(item)

**print**("First Set after removing common element ", first\_set)

**Exercise 8: Iterate a given list and check if a given element exists as a key’s value in a dictionary. If not, delete it from the list**

**Given**:

roll\_number = [47, 64, 69, 37, 76, 83, 95, 97]

sample\_dict = {'Jhon':47, 'Emma':69, 'Kelly':76, 'Jason':97}

**Expected Outcome**:

After removing unwanted elements from list [47, 69, 76, 97]

Ans

roll\_number = [47, 64, 69, 37, 76, 83, 95, 97]

sample\_dict = {'Jhon': 47, 'Emma': 69, 'Kelly': 76, 'Jason': 97}

**print**("List:", roll\_number)

**print**("Dictionary:", sample\_dict)

# create new list

roll\_number[:] = [item **for** item **in** roll\_number **if** item **in** sample\_dict.values()]

**print**("after removing unwanted elements from list:", roll\_number)

**Exercise 9: Get all values from the dictionary and add them to a list but don’t add duplicates**

**Given**:

speed = {'jan': 47, 'feb': 52, 'march': 47, 'April': 44, 'May': 52, 'June': 53, 'july': 54, 'Aug': 44, 'Sept': 54}

**Expected Outcome**:

[47, 52, 44, 53, 54]

speed = {'jan': 47, 'feb': 52, 'march': 47, 'April': 44, 'May': 52, 'June': 53,

'july': 54, 'Aug': 44, 'Sept': 54}

**print**("Dictionary's values - ", speed.values())

speed\_list = **list**()

# iterate dict values

**for** val **in** speed.values():

# check if value not present in a list

**if** val **not** **in** speed\_list:

speed\_list.append(val)

**print**("unique list", speed\_list)

**Exercise 10: Remove duplicates from a list and create a tuple and find the minimum and maximum number**

**Given**:

sample\_list = [87, 45, 41, 65, 94, 41, 99, 94]

**Expected Outcome**:

unique items [87, 45, 41, 65, 99]

tuple (87, 45, 41, 65, 99)

min: 41

max: 99

ans

sample\_list = [87, 52, 44, 53, 54, 87, 52, 53]

print("Original list", sample\_list)

sample\_list = list(set(sample\_list))

print("unique list", sample\_list)

t = tuple(sample\_list)

print("tuple ", t)

print("Minimum number is: ", min(t))

print("Maximum number is: ", max(t))

**Problems on Dictionary**

**Exercise 1: Convert two lists into a dictionary**

Below are the two lists. Write a Python program to convert them into a dictionary in a way that item from list1 is the key and item from list2 is the value

keys = ['Ten', 'Twenty', 'Thirty']

values = [10, 20, 30]

**Expected output:**

{'Ten': 10, 'Twenty': 20, 'Thirty': 30}

Ans

keys = ['Ten', 'Twenty', 'Thirty']

values = [10, 20, 30]

res\_dict = **dict**(**zip**(keys, values))

**print**(res\_dict)

**Exercise 2: Merge two Python dictionaries into one**

dict1 = {'Ten': 10, 'Twenty': 20, 'Thirty': 30}

dict2 = {'Thirty': 30, 'Fourty': 40, 'Fifty': 50}

**Expected output:**

{'Ten': 10, 'Twenty': 20, 'Thirty': 30, 'Fourty': 40, 'Fifty': 50}

dict1 = {'Ten': 10, 'Twenty': 20, 'Thirty': 30}

dict2 = {'Thirty': 30, 'Fourty': 40, 'Fifty': 50}

dict3 = dict1.copy()

dict3.update(dict2)

**print**(dict3)

**Exercise 3: Print the value of key ‘history’ from the below dict**

sampleDict = {

    "class": {

        "student": {

            "name": "Mike",

            "marks": {

                "physics": 70,

                "history": 80

            }

        }

    }

}

**Expected output:**

80

Ans

sampleDict = {

"class": {

"student": {

"name": "Mike",

"marks": {

"physics": 70,

"history": 80

}

}

}

}

**print**(sampleDict['class']['student']['marks']['history'])

**Exercise 4: Initialize dictionary with default values**

In Python, we can initialize the keys with the same values.

**Given**:

employees = ['Kelly', 'Emma']

defaults = {"designation": 'Developer', "salary": 8000}

**Expected output:**

{'Kelly': {'designation': 'Developer', 'salary': 8000}, 'Emma': {'designation': 'Developer', 'salary': 8000}}

ans

employees = ['Kelly', 'Emma']

defaults = {"designation": 'Developer', "salary": 8000}

res = **dict**.fromkeys(employees, defaults)

**print**(res)

# Individual data

**print**(res["Kelly"])

**Exercise 5: Create a dictionary by extracting the keys from a given dictionary**

Write a Python program to create a new dictionary by extracting the mentioned keys from the below dictionary.

**Given dictionary**:

sample\_dict = {

    "name": "Kelly",

    "age": 25,

    "salary": 8000,

    "city": "New york"}

# Keys to extract

keys = ["name", "salary"]

**Expected output:**

{'name': 'Kelly', 'salary': 8000}

Ans

sampleDict = {

"name": "Kelly",

"age":25,

"salary": 8000,

"city": "New york" }

keys = ["name", "salary"]

newDict = {k: sampleDict[k] **for** k **in** keys}

**print**(newDict)

**Exercise 6: Delete a list of keys from a dictionary**

**Given**:

sample\_dict = {

    "name": "Kelly",

    "age": 25,

    "salary": 8000,

    "city": "New york"

}

# Keys to remove

keys = ["name", "salary"]

**Expected output:**

{'city': 'New york', 'age': 25}

Ans

sample\_dict = {

"name": "Kelly",

"age": 25,

"salary": 8000,

"city": "New york"

}

# Keys to remove

keys = ["name", "salary"]

for k in keys:

sample\_dict.pop(k)

print(sample\_dict)

**Exercise 7: Check if a value exists in a dictionary**

We know how to check if the key exists in a dictionary. Sometimes it is required to check if the given value is present.

Write a Python program to check if value 200 exists in the following dictionary.

**Given**:

sample\_dict = {'a': 100, 'b': 200, 'c': 300}

**Expected output:**

200 present in a dict

ANS

sample\_dict = {'a': 100, 'b': 200, 'c': 300}

**if** 200 **in** sample\_dict.values():

**print**('200 present in a dict')

**Exercise 8: Rename key of a dictionary**

Write a program to rename a key city to a location in the following dictionary.

**Given**:

sample\_dict = {

  "name": "Kelly",

  "age":25,

  "salary": 8000,

  "city": "New york"

}

**Expected output:**

{'name': 'Kelly', 'age': 25, 'salary': 8000, 'location': 'New york'}

ANS

sample\_dict = {

"name": "Kelly",

"age": 25,

"salary": 8000,

"city": "New york"

}

sample\_dict['location'] = sample\_dict.pop('city')

**print**(sample\_dict)

**Exercise 9: Get the key of a minimum value from the following dictionary**

sample\_dict = {

  'Physics': 82,

  'Math': 65,

  'history': 75

}

**Expected output:**

Math

**Ans**

sample\_dict = {

'Physics': 82,

'Math': 65,

'history': 75

}

**print**(**min**(sample\_dict, key=sample\_dict.get))

**Exercise 10: Change value of a key in a nested dictionary**

Write a Python program to change Brad’s salary to 8500 in the following dictionary.

**Given**:

sample\_dict = {

    'emp1': {'name': 'Jhon', 'salary': 7500},

    'emp2': {'name': 'Emma', 'salary': 8000},

    'emp3': {'name': 'Brad', 'salary': 500}

}

**Expected output:**

{

   'emp1': {'name': 'Jhon', 'salary': 7500},

   'emp2': {'name': 'Emma', 'salary': 8000},

   'emp3': {'name': 'Brad', 'salary': 8500}

ans

sample\_dict = {

'emp1': {'name': 'Jhon', 'salary': 7500},

'emp2': {'name': 'Emma', 'salary': 8000},

'emp3': {'name': 'Brad', 'salary': 6500}

}

sample\_dict['emp3']['salary'] = 8500

**print**(sample\_dict)