**Python Coding Challenge (18-06-2025)**

**-BY GETSY JACINTH**

Topic: List, Tuple, Dictionary, Set | Total Questions: 10 | Time: 60 minutes

Section A: List (3 Questions):

Section B: Tuple (2 Questions):

Section C: Dictionary (3 Questions):

Section D: Set (2 Questions):

Q1. Write a Python program to remove all duplicates from a list without using the set() function.

Input Example: [1, 2, 2, 3, 4, 4, 5]

Output: [1, 2, 3, 4, 5]

**Code:**

**lst = [1, 2, 2, 3, 4, 4, 5]**

**result = []**

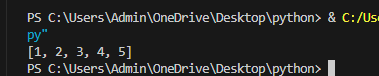
**for x in lst:**

**if x not in result:**

**result.append(x)**

**print(result)**

**output screenshot:**

****

Q2. Given a list of integers, write a program to find the second highest unique number.

Input Example: [12, 5, 9, 21, 21, 3]

Output: 12

**Code:**

**lst = [12, 5, 9, 21, 21, 3]**

**unique = list(set(lst))**

**unique.sort(reverse=True)**

**print(unique[1])**

**output screenshot:**

****

Q3. Rotate a list to the right by k positions.

Input: List = [1, 2, 3, 4, 5], k = 2

Output: [4, 5, 1, 2, 3]

**Code:**

**lst = [1, 2, 3, 4, 5]**

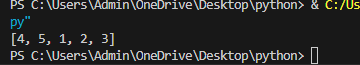
**k = 2**

**k = k % len(lst)**

**rotated = lst[-k:] + lst[:-k]**

**print(rotated)**

**ouput screenshot:**

****

Q4. Write a Python program to multiply the elements of each tuple in a list of tuples and return a new list.

Input: [(2, 4), (3, 5), (4, 6)]

Output: [8, 15, 24]

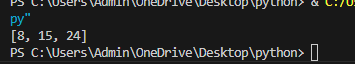
**Code:**

**lst = [(2, 4), (3, 5), (4, 6)]**

**result = [a \* b for a, b in lst]**

**print(result)**

**output screenshot:**

****

Q5. Given a tuple of integers, write a program to count how many times each element occurs.

Input: (1, 2, 2, 3, 1, 4, 2)

Output: {1: 2, 2: 3, 3: 1, 4: 1}

**Code:**

**tup = (1, 2, 2, 3, 1, 4, 2)**

**count = {}**

**for x in tup:**

**count[x] = count.get(x, 0) + 1**

**print(count)**

**output screenshot:**

****

Q6. Write a Python program to count the frequency of each character in a string using a dictionary.

Input: 'banana'

Output: {'b': 1, 'a': 3, 'n': 2}

**Code:**

**s = 'banana'**

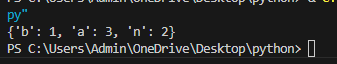
**freq = {}**

**for ch in s:**

**freq[ch] = freq.get(ch, 0) + 1**

**print(freq)**

**output screenshots:**

****

Q7. Merge two dictionaries such that common keys have their values summed.

Input: {'apple': 10, 'banana': 5}, {'banana': 3, 'orange': 7}

Output: {'apple': 10, 'banana': 8, 'orange': 7}

**Code:**

**d1 = {'apple': 10, 'banana': 5}**

**d2 = {'banana': 3, 'orange': 7}**

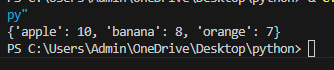
**merged = d1.copy()**

**for k, v in d2.items():**

**merged[k] = merged.get(k, 0) + v**

**print(merged)**

**output screenshot:**

****

Q8. Given a dictionary of student names and their marks, print the name(s) of the student(s) with the highest

marks.

Input: {'Alice': 85, 'Bob': 92, 'Carol': 92}

Output: ['Bob', 'Carol']

**Code:**

**marks = {'Alice': 85, 'Bob': 92, 'Carol': 92}**

**highest = max(marks.values())**

**top\_students = [name for name, mark in marks.items() if mark == highest]**

**print(top\_students)**

**Output screenshot:**

****

Q9. Write a Python program to find all common elements among three lists using set operations.

Input: [1, 2, 3], [2, 3, 4], [3, 2, 5]

Output: {2, 3}

**Code:**

**a = [1, 2, 3]**

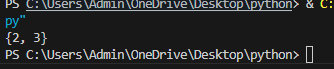
**b = [2, 3, 4]**

**c = [3, 2, 5]**

**common = set(a) & set(b) & set(c)**

**print(common)**

**output screenshot:**

****

Q10. From a sentence entered by the user, extract and display all unique words using a set.

Input: 'this is a test this is fun'

Output: {'this', 'is', 'a', 'test', 'fun'}

**Code:**

**sentence = 'this is a test this is fun'**

**words = sentence.split()**

**unique\_words = set(words)**

**print(unique\_words)**

**output screenshot:**

****