Python for Computer Science and Data Science 1 (CSE 3651)

MINOR ASSIGNMENT-5: DICTIONARIES AND SETS

1. Write a program to accept student name and marks from the user and create a dictionary. Also, display student marks by taking student name as input.

2. Write a program to enter names and percentage of marks in a dictionary and display the information on the screen.

3. Write a program to take a user-input dictionary and print the sum of the values.

4. Make an English-to-French dictionary called **e2f** and print it. Here are your starter words: dog is chien, cat is chat, and walrus is morse.

5. A dictionary which maps country names to Internet top-level domains (TLDs) is given as follows:
   tlds = {'Canada': 'ca', 'United States': 'us', 'Mexico': 'mx'}
   Perform the following tasks and display the results:
   a) Check whether the dictionary contains the key 'Canada'.
   b) Check whether the dictionary contains the key 'France'.
   c) Iterate through the key-value pairs and display them in a two-column format.
   d) Add the key–value pair 'Sweden' and 'sw' (incorrect TLD).
   e) Update the value for the key 'Sweden' to 'se' (correct TLD).
   f) Use dictionary comprehension to reverse the keys and values.

6. For the following dictionary, create lists of its keys, values, and items, and show those lists.
   roman_numerals = {'I': 1, 'II': 2, 'III': 3, 'V': 5}

7. Use dictionary comprehension to create a dictionary of the numbers 1–5 mapped to their cubes.

8. Make a multilevel dictionary called **life**. Use these strings for the topmost keys: 'animals', 'plants', and 'other'. Make the 'animals' key refer to another dictionary with the keys 'cats', 'octopi', and 'emus'. Make the 'cats' key refer to a list of strings with the values 'Henri', 'Grumpy', and 'Lucy'. Make all the other keys refer to empty dictionaries.

9. Write a program to find the number of occurrences of each letter present in a given string., e.g., str='mississippi' $\Rightarrow$ {'m': 1, 'i': 4, 's': 4, 'p': 2}

10. Write a program to find the number of occurrences of each vowel present in a given string, and also print the vowels.

11. Write a function that takes a number as an input parameter and returns the corresponding text in words, e.g., on input 452, the function should return 'Four Five Two'. Use a dictionary for mapping digits to their string representation.

12. Write a program that uses a dictionary to determine and print the number of duplicate words in a sentence. Treat uppercase and lowercase letters the same and assume there is no punctuation in the sentence.

13. Write a function that receives a list of words, then determines and displays in alphabetical order only the unique words. Treat uppercase and lowercase letters as the same. The function should use a set to get the unique words in the list. Test your function with several sentences.

14. Create a program that determines and displays the number of unique characters in a string entered by the user, e.g., Hello, World! has 10 unique characters, while zzz has only one unique character. Use a dictionary or set to solve this problem.

15. Modify a script to play 1,000,000 games of craps. Use two dictionaries, `wins` and `losses`, to track the number of games won and lost for each roll number. Update these dictionaries as the simulation progresses. For example, a key–value pair `4: 50217` in the `wins` dictionary would mean that 50,217 games were won on the 4th roll. At the end of the simulation, display:

    (i) The percentage of games won.

    (ii) The percentage of games lost.

    (iii) The percentage of games resolved on each roll.

    (iv) The cumulative percentage of games resolved up to each roll.

Sample Output

```
Percentage of wins: 50.2%
Percentage of losses: 49.8%
Percentage of wins/losses based on total number of rolls:

Rolls   | % Resolved on this roll | Cumulative % of games resolved
-----------------------------------------------------------------
1       | 30.10%                  | 30.10%
2       | 20.80%                  | 50.90%
...
```

16. Given the sets {10, 20, 30} and {5, 10, 15, 20}, use the mathematical set operators to produce the following sets: a) {30}, b) {5, 15, 30} c) {5, 10, 15, 20, 30} d) {10, 20}

17. Using the sets {'red', 'green', 'blue'}, and {'cyan', 'green', 'blue', 'magenta', 'red'}, display the results of:
    a) comparing the sets using each of the comparison operators.
    b) combining the sets using each of the mathematical set operators.

18. You are given two lists of integers: `list1` and `list2`. Write a Python function `analyze_sets(list1, list2)` that performs the following tasks:

    - Creates two sets, `set1` and `set2`, from `list1` and `list2`.
    - Finds the symmetric difference of `set1` and `set2` (elements that are in either set, but not both).
    - For each element in the symmetric difference:
        - If the element is even, multiply it by 2.
        - If the element is odd, add 5 to it.
    - Return a sorted list of the modified elements.

19. Given a long list of words, write a Python function `unique_pairs(words)` to find all unique pairs of words that:

    - Have no common letters (e.g., `"cat"` and `"dogs"` have no letters in common).

- Each word in the pair should have at least 4 letters.
- Each unique pair should be stored in a set as a tuple in lexicographical order.

The function should return a set of all such unique pairs. **Example**:

```
words = ["apple", "dogs", "cat", "bird", "fish", "zebra", "lion"]
print(unique_pairs(words))
```

Expected Output (Order may vary due to set properties):

```
{("apple", "dogs"), ("apple", "bird"), ("apple", "fish"),
("bird", "dogs"), ("dogs", "fish"), ("dogs", "zebra"),
("dogs", "lion"), ("fish", "zebra"), ("lion", "zebra")}
```

20. Write a Python program to change Mukesh's net worth to 9650 in the following dictionary:

```
sample_dict = {
    'person1': {'name': 'Bezos', 'net worth': 21,880},
    'person2': {'name': 'Elon', 'net worth': 31,570},
    'person3': {'name': 'Mukesh', 'net worth': 965}
}
```