DICTIONARY

DICTIONARY PART-1

1. Write a Python program to check whether a given key already exists in a dictionary.
2. Write a Python program to merge two Python dictionaries.
3. Write a Python program to get the maximum and minimum values of a dictionary.
4. Write a Python program to check if a dictionary is empty or not.
5. Create a nested dictionary representing a person with keys for personal information and address.
6. Create a dictionary representing a person with keys for name, age, and city.
   a. Print the age of the person from the dictionary created in Question 1.
   b. Dictionary Key Existence Check
   c. Check if the key "occupation" exists in the person dictionary.
   d. Remove the key "city" from the person dictionary.
   e. Add the key "occupation" with the value "Engineer" to the person dictionary.
   f. Remove the key "city" from the person dictionary.
   g. Iterate through the person dictionary and print all key-value pairs.

7. Write a program that initializes a dictionary with some key-value pairs and allows the user to search for a key and update its corresponding value.
8. You are given a dictionary called my_dict3 that contains information about students and their grades. Write a program that removes a specific student and their grades from the dictionary.Print the modified dictionary after removing the student.

DICTIONARY PART-2

1. Create a dictionary containing the squares of numbers from 1 to 5.
2. Create a tuple of tuples, where each inner tuple contains a name and an age.
    A. Convert this data into a dictionary where names are keys and ages are values.
    B. Perform various dictionary operations like adding a new entry, updating an age, and deleting an entry.

3. Create a tuple containing a dictionary. Use dictionary unpacking to access and print specific key-value pairs from the dictionary within the tuple.
4. Write a Python program that takes a list of items and prints a dictionary where keys are unique items from the list, and values are their respective frequencies (how many times each item appears in the list).

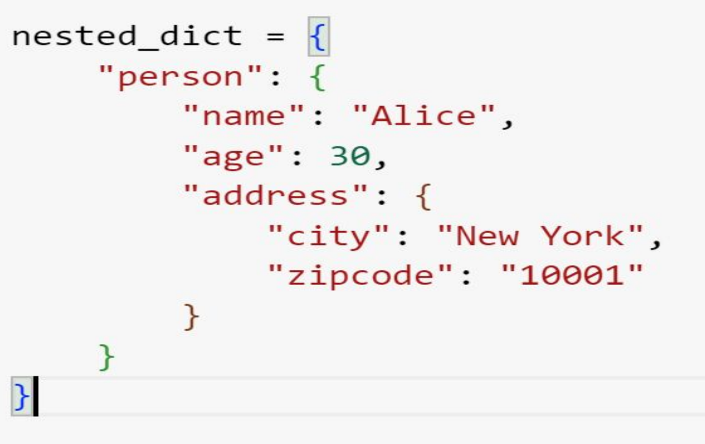
5. Write a Python script to print a dictionary where the keys are numbers between 1 and 15 (both included) and the values are the square of the keys.
   Sample Dictionary
   {1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100, 11: 121, 12: 144, 13: 169, 14: 196, 15: 225}

6. Sort and print the keys of a dictionary in alphabetical order.
7. Count the frequency of each character in a given string and store the results in a dictionary.
8. Calculate the sum of all values in a dictionary of numbers

DICTIONARY PART-3

1. Write a Python program to print all distinct values in a dictionary.
2. Find and print the key in a dictionary with the maximum value.
3. Swap the keys and values of a dictionary.
4. Create a dictionary from two lists, one for keys and another for values.
5. Iterate through a nested dictionary and print all key-value pairs.
6. Filter a dictionary to exclude specific keys.
7. Update the value of a nested dictionary key.

   Update city in the address



```
nested_dict = {
    "person": {
        "name": "Alice",
        "age": 30,
        "address": {
            "city": "New York",
            "zipcode": "10001"
        }
    }
}
```

DICTIONARY PART-4

1. Sort a dictionary by its keys and print the sorted key-value pairs.
2. Sort a dictionary by its values in descending order and print the sorted key-value pairs.
3. Find the intersection of two dictionaries.
4. Find and print the key in a dictionary with the minimum value.
5. Convert a dictionary to a string.
6. You are given a dictionary called my_dict1 that contains information about students and their ages. Write a program that prints the total number of students in the dictionary.

7. Create a dictionary where keys are from a list and values are a default value (e.g., 0).

8. Modify the values in a dictionary by applying a specific operation (e.g., doubling the values).

9. Check if a specific key exists in a nested dictionary.

10. Reverse the dictionary by swapping keys and values.

DICTIONARY PART-5

1. Write a program that takes key-value pairs as input and stores them in a dictionary. Continue this process until the user enters "done". Then, print the resulting dictionary.
2. Write a program that takes a key as input and checks if it exists in the dictionary. If the key exists, print its corresponding value; otherwise, print "Key not found".
3. Write a program that takes a key as input and deletes it from the dictionary. Continue this process until the user enters "done".
4. Create a program for managing a shop's inventory. Implement the following functionalities using a dictionary.Continue this process until the user enters "exit".
   :
   A. Add a new product to the inventory.
   B. Update the quantity of an existing product.
   C. Display the inventory.

5. Write a Python program that takes a list of words and counts the frequency of each word. Print the word and its frequency in a dictionary format.
6. You have a dictionary called inventory that contains information about products and their quantities. Write a program to calculate and print the total quantity of each product in the inventory.
7. Write a program that takes student names and their grades as input, stores them in a dictionary, and then calculates and prints the average grade.

DICTIONARY PART-6

1. Create a program for a shop where each product has a price. Implement the following functionalities using a dictionary and while loop:
    A. Add a new product with price to the shop.
    B. Record a sale by specifying the product and quantity sold.
    C. Calculate and apply a discount of 10% for total sales above a certain amount (e.g., $100).
    D. Calculate and display the total sales after applying the discount.
2. Write a program that implements a simple quiz game using a dictionary to store questions and answers. The program should continue asking questions until the user decides to exit.
3. Write a program that allows the user to manage a phonebook. They should be able to add, search, and delete contacts.
4. Enhance the simple quiz game to include a scoring system. Assign scores for correct answers and display the total score at the end.
5. Write a program that allows the user to track their daily expenses. Allow them to add expenses with categories and calculate the total spending.
6. You have a dictionary called shopping_list that contains items as keys and their quantities as values. Write a program that simulates a shopping trip by repeatedly asking the user to enter an item they have purchased. After each input, update the shopping_list by reducing the quantity of the purchased item by 1. If the quantity becomes 0, remove the item from the shopping_list using the clear() method. Continue this process until the user indicates they are done shopping by entering "done". Finally, print the updated shopping_list.