Dictionary Tasks

Task 1

Python Program to Add a Key-Value Pair to the Dictionary

Problem Solution:

1. Take a key-value pair from the user and store it in separate variables.

2. Declare a dictionary and initialize it to an empty dictionary.

3. Use the update() function to add the key-value pair to the dictionary.

4. Print the final dictionary.

5. Exit.

Runtime Test Cases

```
Case 1:
Enter the key (int) to be added:12
Enter the value for the key to be added:34
Updated dictionary is:
{12: 34}

Case 2:
Enter the key (int) to be added:34
Enter the value for the key to be added:29
Updated dictionary is:
{34: 29}
```

Task 2

Python Program to Concatenate Two Dictionaries Into One

Problem Solution

1. Declare and initialize two dictionaries with some key-value pairs

2. Use the update() function to add the key-value pair from the second dictionary to the first dictionary.

3. Print the final dictionary.

4. Exit.

Runtime Test Cases

```
Case 1:
Concatenated dictionary is:
{'A': 1, 'C': 3, 'B': 2}
```

Task 3

Python Program to Check if a Given Key Exists in a Dictionary or Not

Problem Solution

1. Declare and initialize a dictionary to have some key-value pairs.

2. Take a key from the user and store it in a variable.

3. Using an if statement and the in operator, check if the key is present in the dictionary using the dictionary.keys() method.

4. If it is present, print the value of the key.

5. If it isn't present, display that the key isn't present in the dictionary.

6. Exit.

Runtime Test Cases

```
Case 1:
Enter key to check:A
Key is present and value of the key is:
1

Case 2:
Enter key to check:F
Key isn't present!
```

Task 4

Python Program to Generate a Dictionary that Contains Numbers (between 1 and n) in the Form (x,x*x).

Problem Solution

1. Take a number from the user and store it in a separate variable.

2. Declare a dictionary and using dictionary comprehension initialize it to values keeping the number between 1 to n as the key and the square of the number as their values.

3. Print the final dictionary.

4. Exit.

```
Case 1:
Enter a number:5
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25}

Case 2:
```

```
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100, 11: 121, 12: 14
4, 13: 169, 14: 196, 15: 225, 16: 256, 17: 289, 18: 324, 19: 361}
```

Task 5

Python Program to Sum All the Items in a Dictionary

Problem Solution

1. Declare and initialize a dictionary to have some key-value pairs.

2. Find the sum of all the values in the dictionary.

3. Print the total sum.

4. Exit.

Runtime Test Cases

```
Case 1:
Total sum of values in the dictionary:
125
```

Task 6

Python Program to Multiply All the Items in a Dictionary

Problem Solution

1. Declare and initialize a dictionary to have some key-value pairs.

2. Initialize a variable that should contain the total multiplied value to 1.

3. Use the for loop to traverse through the values of the dictionary.

4. Then multiply all the values in the dictionary against each other.

5. Print the total multiplied value.

6. Exit.

Runtime Test Cases

```
Case 1:
The multiplication of all items is: 23900
```

Task 7

Python Program to Remove the Given Key from a Dictionary

Problem Solution

1. Declare and initialize a dictionary to have some key-value pairs.

2. Take a key from the user and store it in a variable.

3. Using an if statement and the in operator, check if the key is present in the dictionary.

4. If it is present, delete the key-value pair.

5. If it isn't present, print that the key isn't found and exit the program.

6. Exit.

Runtime Test Cases

```
Case 1:
Initial dictionary
{'a': 1, 'c': 3, 'b': 2, 'd': 4}
Enter the key to delete(a-d):c
Updated dictionary
{'a': 1, 'b': 2, 'd': 4}

Case 2:
Initial dictionary
{'a': 1, 'c': 3, 'b': 2, 'd': 4}
Enter the key to delete(a-d):g
Key not found!
```

Task 8

Python Program to Map Two Lists into a Dictionary

Problem Solution

1. Declare two empty lists and initialize them to an empty list.

3. Consider a for loop to accept values for the two lists.

4. Take the number of elements in the list and store it in a variable.

5. Accept the values into the list using another for loop and insert into the list.

6. Repeat 4 and 5 for the values list also.

7. Zip the two lists and use dict() to convert it into a dictionary.

8. Print the dictionary.

9. Exit.

Runtime Test Cases

```
Case 1:
Enter number of elements for dictionary:3
For keys:
Enter element1:1
Enter element2:2
Enter element3:3
For values:
Enter element1:1
Enter element2:4
Enter element3:9
The dictionary is:
{1: 1, 2: 4, 3: 9}

Case 2:
Enter number of elements for dictionary:2
For keys:
Enter element1:23
Enter element2:46
For values:
Enter element1:69
Enter element2:138
The dictionary is:
{46: 138, 23: 69}
```

Task 9

Python Program to Count the Frequency of Words Appearing in a String Using a Dictionary

Problem Solution

1. Enter a string and store it in a variable.

2. Declare a list variable and initialize it to an empty list.

3. Split the string into words and store it in the list.

4. Count the frequency of each word and store it in another list.

5. Using the zip() function, merge the lists containing the words and the word counts into a dictionary.

3. Print the final dictionary.

4. Exit.

Runtime Test Cases

```
Case 1:
Enter string:hello world program world test
{'test': 1, 'world': 2, 'program': 1, 'hello': 1}
```

```
Case 2:
Enter string:orange banana apple apple orange pineapple
{'orange': 2, 'pineapple': 1, 'banana': 1, 'apple': 2}
```

Task 10

Python Program to Create a Dictionary with Key as First Character and Value as Words Starting with that Character

Problem Solution

1. Enter a string and store it in a variable.

2. Declare an empty dictionary.

3. Split the string into words and store it in a list.

4. Using a for loop and if statement check if the word already present as a key in the dictionary.

5. If it is not present, initialize the letter of the word as the key and the word as the value and append it to a sublist created in the list.

6. If it is present, add the word as the value to the corresponding sublist.

7. Print the final dictionary.

8. Exit.

Runtime Test Cases

```
Case 1:
Enter string:Hello world this is a test string sanfoundry
('a', ':', ['a'])
('i', ':', ['is'])
('H', ':', ['Hello'])
('s', ':', ['sanfoundry', 'string'])
('t', ':', ['test', 'this'])
('w', ':', ['world'])

Case 2:
Enter string:python is my most favourite programming language in the entire world
('e', ':', ['entire'])
('f', ':', ['favourite'])
('i', ':', ['in', 'is'])
('m', ':', ['most', 'my'])
('l', ':', ['language'])
('p', ':', ['programming', 'python'])
('t', ':', ['the'])
('w', ':', ['world'])
```