

1. Write a Python program to sort Counter by value.

Sample Input	Sample Output
{'Math':81, 'Physics':83, 'Chemistry':87}	[('Chemistry', 87), ('Physics', 83), ('Math', 81)]

2. Write a Python program to combine values in a list of dictionaries.

Sample Input	Sample Output
[{'item': 'item1', 'amount': 400}, {'item': 'item2', 'amount': 300}, {'item': 'item1', 'amount': 750}]	{'item1': 1150, 'item2': 300}

3. Write a Python program to create a dictionary from a string.
Note: Track the count of the letters from the string.

Sample Input	Sample Output
'w3resource'	{'w': 1, '3': 1, 'r': 2, 'e': 2, 's': 1, 'o': 1, 'u': 1, 'c': 1}

4. Create a dictionary by extracting the keys from a given dictionary

Sample Input	Sample Output
<pre>sample_dict = { "name": "Kelly", "age": 25, "salary": 8000, "city": "New york"} Keys to extract keys = ["name", "salary"]</pre>	{'name': 'Kelly', 'salary': 8000}

--	--

5. Write a Python program to convert a list to a list of dictionaries.

Sample Input	Sample Output
["Black", "Red", "Maroon", "Yellow"], ["#000000", "#FF0000", "#800000", "#FFFF00"]	{'color_name': 'Black', 'color_code': '#000000'}, {'color_name': 'Red', 'color_code': '#FF0000'}, {'color_name': 'Maroon', 'color_code': '#800000'}, {'color_name': 'Yellow', 'color_code': '#FFFF00'}

6. Write a Python program that takes a sentence as input and calculates the frequency of each word. The program should output a dictionary where keys are unique words, and values are the frequencies of those words.

Sample Input	Sample Output
the quick brown fox jumps over the lazy dog the lazy dog barks"	{'the': 3, 'quick': 1, 'brown': 1, 'fox': 1, 'jumps': 1, 'over': 1, 'lazy': 2, 'dog': 2, 'barks': 1}

7. Given a list of integers, write a Python program to create a dictionary where the keys are the integers, and the values are their squares, but only for odd integers.

Sample Input	Sample Output
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]	{1: 1, 3: 9, 5: 25, 7: 49, 9: 81}

8. UIU database has a dictionary containing student data. Each key is a student's name, and the corresponding value is a dictionary with information such as the student's name, major, CGPA (Cumulative Grade Point Average), and completed credits. Imagine you are a software engineer working at UIU, you are assigned to write a Python program that will transform this dictionary filled with redundancy into an optimized one.

Now, Create a new dictionary with the same student names as keys and values generated by extracting each student's "cgpa" and "completed_credits" information and arrange them sorted based on their CGPA.

Sample Input	Sample Output
Original student data: <pre>{ 'Emma': {'name': 'Emma', 'major': 'Computer Science', 'cgpa': 3.8, 'completed_credits': 90}, 'Daniel': {'name': 'Daniel', 'major': 'Electrical Engineering', 'cgpa': 3.5, 'completed_credits': 75}, 'Sophia': {'name': 'Sophia', 'major': 'Mechanical Engineering', 'cgpa': 3.2, 'completed_credits': 60} }</pre>	Transformed student data: <pre>{ 'Emma': {'cgpa': 3.8, 'completed_credits': 90}, 'Daniel': {'cgpa': 3.5, 'completed_credits': 75}, 'Sophia': {'cgpa': 3.2, 'completed_credits': 60} }</pre>

9. In the upcoming year 2024, a tech company is looking to update its HR records implemented in Python dictionaries. The company plans to acknowledge and reward senior employees aged over 55 with a bonus of 10,000 BDT as they approach retirement. Additionally, a bonus of 5,000 BDT will be granted to employees whose performance exceeds 95%, and those with a performance below 60% will be terminated from the records and will receive no bonus.

Compute the total number of employees in the company in 2024, the overall bonus amount required and show the updated dictionary.

Sample Input	Sample Output
<pre>company_hr_register = { 101: {'name': 'Alice', 'age': 35, 'performance': 90, 'salary': 50000}, 102: {'name': 'Bob', 'age': 58, 'performance': 98, 'salary': 70000}, 103: {'name': 'Charlie', 'age': 45, 'performance': 85, 'salary': 60000}, 104: {'name': 'David', 'age': 60, 'performance': 75, 'salary': 55000}, 105: {'name': 'Eve', 'age': 28, 'performance': 92, 'salary': 48000},</pre>	<pre>total_employees = 2 total_bonus_amount = 15000 updated_company_hr_register = { 102: {'name': 'Bob'}, 107: {'name': 'Grace'}, }</pre>

106: {'name': 'Frank', 'age': 50, 'performance': 55, 'salary': 52000}, 107: {'name': 'Grace', 'age': 62, 'performance': 97, 'salary': 75000}, }	
---	--

10. A mathematician Meera has found interest in prime numbers. She discovered a really fascinating concept called **Prime Powers**. Prime powers are numbers obtained by multiplying a prime number by itself a certain number of times. For example, 2^3 is a prime power because it is the result of **multiplying the prime number 2** by itself **three times** which is also a prime number ($2 * 2 * 2 = 8$). Similarly, 2^2 , 2^5 all are prime powers.

She is trying to create a Python program that takes an integer **n** as input and generates a dictionary where the **keys are prime numbers** and the **values are lists containing all the prime powers** from 2 to n. However, she does not want to include the prime powers that are perfect squares and does not want to display the keys that have no values. Hint: make sure the value of prime powers are under 100.

Sample Input	Sample Output
Input =10	Output: { 2: [8, 32], 3: [27], }

Explanation: Here, no other keys have a prime power whose value is less than 100 and is not a perfect square