PW SKILLS

Assignment

Assignments: Data Structures in Python

1. Write a code to reverse a string.
2. Write a code to count the number of vowels in a string.
3. Write a code to check if a given string is a palindrome or not.
4. Write a code to check if two given strings are anagrams of each other.
5. Write a code to find all occurrences of a given substring within another string.
6. Write a code to perform basic string compression using the counts of repeated characters.
7. Write a code to determine if a string has all unique characters.
8. Write a code to convert a given string to uppercase or lowercase.
9. Write a code to count the number of words in a string.
10. Write a code to concatenate two strings without using the + operator.
11. Write a code to remove all occurrences of a specific element from a list.
12. Implement a code to find the second largest number in a given list of integers.
13. Create a code to count the occurrences of each element in a list and return a dictionary with elements as keys and their counts as values.
14. Write a code to reverse a list in-place without using any built-in reverse functions.
15. Implement a code to find and remove duplicates from a list while preserving the original order of elements.
16. Create a code to check if a given list is sorted (either in ascending or descending order) or not.
17. Write a code to merge two sorted lists into a single sorted list.
18. Implement a code to find the intersection of two given lists.
19. Create a code to find the union of two lists without duplicates.
20. Write a code to shuffle a given list randomly without using any built-in shuffle functions.
21. Write a code that takes two tuples as input and returns a new tuple containing elements that are common to both input tuples.
22. Create a code that prompts the user to enter two sets of integers separated by commas. Then, print the intersection of these two sets.
23. Write a code to concatenate two tuples. The function should take two tuples as input and return a new tuple containing elements from both input tuples.

24. Develop a code that prompts the user to input two sets of strings. Then, print the elements that are present in the first set but not in the second set.

25. Create a code that takes a tuple and two integers as input. The function should return a new tuple containing elements from the original tuple within the specified range of indices.

26. Write a code that prompts the user to input two sets of characters. Then, print the union of these two sets.

27. Develop a code that takes a tuple of integers as input. The function should return the maximum and minimum values from the tuple using tuple unpacking.

28. Create a code that defines two sets of integers. Then, print the union, intersection, and difference of these two sets.

29. Write a code that takes a tuple and an element as input. The function should return the count of occurrences of the given element in the tuple.

30. Develop a code that prompts the user to input two sets of strings. Then, print the symmetric difference of these two sets.

31. Write a code that takes a list of words as input and returns a dictionary where the keys are unique words and the values are the frequencies of those words in the input list.

32. Write a code that takes two dictionaries as input and merges them into a single dictionary. If there are common keys, the values should be added together.

33. Write a code to access a value in a nested dictionary. The function should take the dictionary and a list of keys as input, and return the corresponding value. If any of the keys do not exist in the dictionary, the function should return None.

34. Write a code that takes a dictionary as input and returns a sorted version of it based on the values. You can choose whether to sort in ascending or descending order.

35. Write a code that inverts a dictionary, swapping keys and values. Ensure that the inverted dictionary correctly handles cases where multiple keys have the same value by storing the keys as a list in the inverted dictionary.