Python Programming Practice Problems

Easy Level (1-40)

Lists

1. Create a list of numbers and print each element.

2. Find the length of a list without using len().

3. Reverse a list using slicing.

4. Check if an element exists in a list.

5. Append elements to a list.

6. Remove duplicates from a list.

7. Sort a list in ascending order.

8. Find the largest element in a list using a loop.

9. Remove an element from a list using index.

10. Concatenate two lists.

Tuples

11. Create a tuple of numbers and print each element.

12. Convert a tuple to a list and vice versa.

13. Unpack a tuple into variables.

14. Find the length of a tuple using len().

15. Check if an element exists in a tuple.

16. Concatenate two tuples.

17. Slice a tuple to get a portion of it.

18. Find the index of an element in a tuple.

19. Count occurrences of an element in a tuple.

20. Reverse a tuple.

Sets

21. Create a set of numbers and print each element.

22. Add an element to a set.

23. Remove an element from a set.

24. Find the union of two sets.

25. Find the intersection of two sets.

26. Check if a set is a subset of another.

27. Remove duplicates from a list using a set.

28. Find the difference between two sets.

29. Find the symmetric difference between two sets.

30. Clear all elements from a set.

Dictionaries

31. Create a dictionary with key-value pairs.

32. Access the value associated with a key.

33. Add a new key-value pair to a dictionary.

34. Remove a key-value pair from a dictionary.

35. Check if a key exists in a dictionary.

36. Merge two dictionaries into one.

37. Get all keys from a dictionary.

38. Get all values from a dictionary.

39. Find the length of a dictionary.

40. Update the value of a specific key in a dictionary.

Medium Level (41-80)

Lists

41. Find the second largest element in a list.

42. Count the frequency of elements in a list.

43. Split a list into chunks of a specified size.

44. Rotate a list by a given number of positions.

45. Flatten a nested list.

46. Generate all possible pairs of elements from two lists.

47. Find the common elements between two lists.

48. Find the difference between two lists.

49. Convert a list of tuples into a dictionary.

50. Multiply all elements in a list by a constant.

Tuples

51. Find the maximum and minimum values in a tuple.

52. Convert a list of tuples into separate lists.

53. Sort a tuple by its second element.

54. Find the index of an element in a tuple using a loop.

55. Swap the first and last element in a tuple.

56. Merge two sorted tuples into one sorted tuple.

57. Count the occurrences of a tuple inside a list.

58. Convert a string into a tuple of characters.

59. Find the length of each tuple in a list of tuples.

60. Zip two tuples into a list of tuples.

Sets

61. Check if two sets are disjoint.

62. Find the Cartesian product of two sets.

63. Check if all elements of one set are in another.

64. Find the set of all subsets of a given set.

65. Create a set of unique tuples from a list of tuples.

66. Check if a set is empty.

67. Find the powerset of a given set.

68. Update a set with another set?s elements.

69. Convert a list to a set and back to a list.

70. Check if a set is frozen.

Dictionaries

71. Sort a dictionary by its keys.

72. Sort a dictionary by its values.

73. Invert a dictionary (swap keys and values).

74. Find the maximum value in a dictionary.

75. Group a list of dictionaries by a common key.

76. Merge two dictionaries, summing values for common keys.

77. Check if two dictionaries are equal.

78. Count the frequency of words in a given string using a dictionary.

79. Convert a list of key-value pairs into a dictionary.

80. Remove duplicate values from a dictionary.

Hard Level (81-100)

Lists

81. Find all permutations of a list.

82. Implement binary search on a sorted list.

83. Find the longest increasing subsequence in a list.

84. Remove all occurrences of a sublist from a list.

85. Create a list comprehension to find prime numbers in a range.

86. Find all unique combinations of elements in a list that sum to a target value.

87. Split a list into two sublists where the sum of elements is nearly equal.

88. Write a custom sorting function for a list of lists based on multiple criteria.

89. Find the most common subsequence in a list of integers.

90. Implement a list-based stack with push, pop, and peek operations.

Tuples

91. Find all unique pairs of tuples that satisfy a condition (e.g., sum equals a target).

92. Create a tuple from a nested list by extracting specific elements.

93. Find the longest common prefix among a list of tuples.

94. Use a tuple as a key in a dictionary and solve for a specific value.

95. Write a function to rotate tuples by k positions.

96. Find the top n largest elements from a tuple without converting it to a list.

97. Generate a tuple with the first N elements of the Fibonacci sequence.

98. Find common prefixes among a list of tuples.

99. Write a function to replace specific elements in a tuple with new values.

100. Convert a nested tuple into a flat list.