

GoLang Fullstack Practice: 200 Coding Problems

Problem 1: Sum of Two Numbers (Easy)

Write a function that takes two integers and returns their sum.

Input: a = 5, b = 3

Output: 8

Problem 2: Check Even or Odd (Easy)

Determine whether a given integer is even or odd.

Input: n = 7

Output: Odd

Problem 3: Maximum of Three Numbers (Easy)

Write a function that returns the largest of three given numbers.

Input: a = 10, b = 25, c = 7

Output: 25

Problem 4: Factorial of a Number (Easy)

Compute the factorial of a non-negative integer n. Use recursion.

Input: n = 5

Output: 120

Problem 5: Fibonacci Sequence (Easy)

Print the first N numbers in the Fibonacci sequence.

Input: N = 7

Output: 0 1 1 2 3 5 8

Problem 6: Sample Challenge 6 (Easy)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [5, 6, 7]

Output: Example result for input [5, 6, 7]

Problem 7: Sample Challenge 7 (Easy)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [6, 7, 8]

Output: Example result for input [6, 7, 8]

Problem 8: Sample Challenge 8 (Easy)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [7, 8, 9]

Output: Example result for input [7, 8, 9]

Problem 9: Sample Challenge 9 (Easy)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [8, 9, 10]

Output: Example result for input [8, 9, 10]

Problem 10: Sample Challenge 10 (Easy)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [9, 10, 11]

Output: Example result for input [9, 10, 11]

Problem 11: Sample Challenge 11 (Easy)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [10, 11, 12]

Output: Example result for input [10, 11, 12]

Problem 12: Sample Challenge 12 (Easy)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [11, 12, 13]

Output: Example result for input [11, 12, 13]

Problem 13: Sample Challenge 13 (Easy)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [12, 13, 14]

Output: Example result for input [12, 13, 14]

Problem 14: Sample Challenge 14 (Easy)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [13, 14, 15]

Output: Example result for input [13, 14, 15]

Problem 15: Sample Challenge 15 (Easy)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [14, 15, 16]

Output: Example result for input [14, 15, 16]

Problem 16: Sample Challenge 16 (Easy)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [15, 16, 17]

Output: Example result for input [15, 16, 17]

Problem 17: Sample Challenge 17 (Easy)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [16, 17, 18]

Output: Example result for input [16, 17, 18]

Problem 18: Sample Challenge 18 (Easy)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [17, 18, 19]

Output: Example result for input [17, 18, 19]

Problem 19: Sample Challenge 19 (Easy)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [18, 19, 20]

Output: Example result for input [18, 19, 20]

Problem 20: Sample Challenge 20 (Easy)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [19, 20, 21]

Output: Example result for input [19, 20, 21]

Problem 21: Sample Challenge 21 (Easy)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [20, 21, 22]

Output: Example result for input [20, 21, 22]

Problem 22: Sample Challenge 22 (Easy)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [21, 22, 23]

Output: Example result for input [21, 22, 23]

Problem 23: Sample Challenge 23 (Easy)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [22, 23, 24]

Output: Example result for input [22, 23, 24]

Problem 24: Sample Challenge 24 (Easy)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [23, 24, 25]

Output: Example result for input [23, 24, 25]

Problem 25: Sample Challenge 25 (Easy)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [24, 25, 26]

Output: Example result for input [24, 25, 26]

Problem 26: Sample Challenge 26 (Medium)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [25, 26, 27]

Output: Example result for input [25, 26, 27]

Problem 27: Sample Challenge 27 (Medium)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [26, 27, 28]

Output: Example result for input [26, 27, 28]

Problem 28: Sample Challenge 28 (Medium)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [27, 28, 29]

Output: Example result for input [27, 28, 29]

Problem 29: Sample Challenge 29 (Medium)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [28, 29, 30]

Output: Example result for input [28, 29, 30]

Problem 30: Sample Challenge 30 (Medium)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [29, 30, 31]

Output: Example result for input [29, 30, 31]

Problem 31: Sample Challenge 31 (Medium)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [30, 31, 32]

Output: Example result for input [30, 31, 32]

Problem 32: Sample Challenge 32 (Medium)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [31, 32, 33]

Output: Example result for input [31, 32, 33]

Problem 33: Sample Challenge 33 (Medium)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [32, 33, 34]

Output: Example result for input [32, 33, 34]

Problem 34: Sample Challenge 34 (Medium)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [33, 34, 35]

Output: Example result for input [33, 34, 35]

Problem 35: Sample Challenge 35 (Medium)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [34, 35, 36]

Output: Example result for input [34, 35, 36]

Problem 36: Sample Challenge 36 (Medium)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [35, 36, 37]

Output: Example result for input [35, 36, 37]

Problem 37: Sample Challenge 37 (Medium)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [36, 37, 38]

Output: Example result for input [36, 37, 38]

Problem 38: Sample Challenge 38 (Medium)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [37, 38, 39]

Output: Example result for input [37, 38, 39]

Problem 39: Sample Challenge 39 (Medium)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [38, 39, 40]

Output: Example result for input [38, 39, 40]

Problem 40: Sample Challenge 40 (Medium)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming

elements based on specific rules.

Input: nums = [39, 40, 41]

Output: Example result for input [39, 40, 41]

Problem 41: Sample Challenge 41 (Medium)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [40, 41, 42]

Output: Example result for input [40, 41, 42]

Problem 42: Sample Challenge 42 (Medium)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [41, 42, 43]

Output: Example result for input [41, 42, 43]

Problem 43: Sample Challenge 43 (Medium)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [42, 43, 44]

Output: Example result for input [42, 43, 44]

Problem 44: Sample Challenge 44 (Medium)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [43, 44, 45]

Output: Example result for input [43, 44, 45]

Problem 45: Sample Challenge 45 (Medium)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [44, 45, 46]

Output: Example result for input [44, 45, 46]

Problem 46: Sample Challenge 46 (Medium)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [45, 46, 47]

Output: Example result for input [45, 46, 47]

Problem 47: Sample Challenge 47 (Medium)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [46, 47, 48]

Output: Example result for input [46, 47, 48]

Problem 48: Sample Challenge 48 (Medium)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [47, 48, 49]

Output: Example result for input [47, 48, 49]

Problem 49: Sample Challenge 49 (Medium)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [48, 49, 50]

Output: Example result for input [48, 49, 50]

Problem 50: Sample Challenge 50 (Medium)

Write a GoLang function to process a list of integers. Task could involve filtering, summing, or transforming elements based on specific rules.

Input: nums = [49, 50, 51]

Output: Example result for input [49, 50, 51]

Problem 51: Week 2 Challenge 1 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_51

Output: expected_output_51

Problem 52: Week 2 Challenge 2 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_52

Output: expected_output_52

Problem 53: Week 2 Challenge 3 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_53

Output: expected_output_53

Problem 54: Week 2 Challenge 4 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_54

Output: expected_output_54

Problem 55: Week 2 Challenge 5 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_55

Output: expected_output_55

Problem 56: Week 2 Challenge 6 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_56

Output: expected_output_56

Problem 57: Week 2 Challenge 7 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_57

Output: expected_output_57

Problem 58: Week 2 Challenge 8 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_58

Output: expected_output_58

Problem 59: Week 2 Challenge 9 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_59

Output: expected_output_59

Problem 60: Week 2 Challenge 10 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_60

Output: expected_output_60

Problem 61: Week 2 Challenge 11 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_61

Output: expected_output_61

Problem 62: Week 2 Challenge 12 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_62

Output: expected_output_62

Problem 63: Week 2 Challenge 13 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_63

Output: expected_output_63

Problem 64: Week 2 Challenge 14 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_64

Output: expected_output_64

Problem 65: Week 2 Challenge 15 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_65

Output: expected_output_65

Problem 66: Week 2 Challenge 16 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_66

Output: expected_output_66

Problem 67: Week 2 Challenge 17 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_67

Output: expected_output_67

Problem 68: Week 2 Challenge 18 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_68

Output: expected_output_68

Problem 69: Week 2 Challenge 19 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_69

Output: expected_output_69

Problem 70: Week 2 Challenge 20 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_70

Output: expected_output_70

Problem 71: Week 2 Challenge 21 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_71

Output: expected_output_71

Problem 72: Week 2 Challenge 22 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_72

Output: expected_output_72

Problem 73: Week 2 Challenge 23 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_73

Output: expected_output_73

Problem 74: Week 2 Challenge 24 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_74

Output: expected_output_74

Problem 75: Week 2 Challenge 25 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_75

Output: expected_output_75

Problem 76: Week 2 Challenge 26 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_76

Output: expected_output_76

Problem 77: Week 2 Challenge 27 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_77

Output: expected_output_77

Problem 78: Week 2 Challenge 28 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_78

Output: expected_output_78

Problem 79: Week 2 Challenge 29 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to

Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_79

Output: expected_output_79

Problem 80: Week 2 Challenge 30 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_80

Output: expected_output_80

Problem 81: Week 2 Challenge 31 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_81

Output: expected_output_81

Problem 82: Week 2 Challenge 32 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_82

Output: expected_output_82

Problem 83: Week 2 Challenge 33 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_83

Output: expected_output_83

Problem 84: Week 2 Challenge 34 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_84

Output: expected_output_84

Problem 85: Week 2 Challenge 35 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_85

Output: expected_output_85

Problem 86: Week 2 Challenge 36 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_86

Output: expected_output_86

Problem 87: Week 2 Challenge 37 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_87

Output: expected_output_87

Problem 88: Week 2 Challenge 38 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_88

Output: expected_output_88

Problem 89: Week 2 Challenge 39 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_89

Output: expected_output_89

Problem 90: Week 2 Challenge 40 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_90

Output: expected_output_90

Problem 91: Week 2 Challenge 41 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_91

Output: expected_output_91

Problem 92: Week 2 Challenge 42 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_92

Output: expected_output_92

Problem 93: Week 2 Challenge 43 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_93

Output: expected_output_93

Problem 94: Week 2 Challenge 44 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_94

Output: expected_output_94

Problem 95: Week 2 Challenge 45 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_95

Output: expected_output_95

Problem 96: Week 2 Challenge 46 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_96

Output: expected_output_96

Problem 97: Week 2 Challenge 47 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_97

Output: expected_output_97

Problem 98: Week 2 Challenge 48 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_98

Output: expected_output_98

Problem 99: Week 2 Challenge 49 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_99

Output: expected_output_99

Problem 100: Week 2 Challenge 50 (Easy)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 2 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_100

Output: expected_output_100

Problem 101: Week 3 Challenge 1 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_101

Output: expected_output_101

Problem 102: Week 3 Challenge 2 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_102

Output: expected_output_102

Problem 103: Week 3 Challenge 3 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_103

Output: expected_output_103

Problem 104: Week 3 Challenge 4 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_104

Output: expected_output_104

Problem 105: Week 3 Challenge 5 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_105

Output: expected_output_105

Problem 106: Week 3 Challenge 6 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_106

Output: expected_output_106

Problem 107: Week 3 Challenge 7 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_107

Output: expected_output_107

Problem 108: Week 3 Challenge 8 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_108

Output: expected_output_108

Problem 109: Week 3 Challenge 9 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_109

Output: expected_output_109

Problem 110: Week 3 Challenge 10 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_110

Output: expected_output_110

Problem 111: Week 3 Challenge 11 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_111

Output: expected_output_111

Problem 112: Week 3 Challenge 12 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_112

Output: expected_output_112

Problem 113: Week 3 Challenge 13 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_113

Output: expected_output_113

Problem 114: Week 3 Challenge 14 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_114

Output: expected_output_114

Problem 115: Week 3 Challenge 15 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_115

Output: expected_output_115

Problem 116: Week 3 Challenge 16 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_116

Output: expected_output_116

Problem 117: Week 3 Challenge 17 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_117

Output: expected_output_117

Problem 118: Week 3 Challenge 18 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to

Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_118

Output: expected_output_118

Problem 119: Week 3 Challenge 19 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_119

Output: expected_output_119

Problem 120: Week 3 Challenge 20 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_120

Output: expected_output_120

Problem 121: Week 3 Challenge 21 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_121

Output: expected_output_121

Problem 122: Week 3 Challenge 22 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_122

Output: expected_output_122

Problem 123: Week 3 Challenge 23 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_123

Output: expected_output_123

Problem 124: Week 3 Challenge 24 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_124

Output: expected_output_124

Problem 125: Week 3 Challenge 25 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_125

Output: expected_output_125

Problem 126: Week 3 Challenge 26 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_126

Output: expected_output_126

Problem 127: Week 3 Challenge 27 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_127

Output: expected_output_127

Problem 128: Week 3 Challenge 28 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_128

Output: expected_output_128

Problem 129: Week 3 Challenge 29 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_129

Output: expected_output_129

Problem 130: Week 3 Challenge 30 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_130

Output: expected_output_130

Problem 131: Week 3 Challenge 31 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_131

Output: expected_output_131

Problem 132: Week 3 Challenge 32 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_132

Output: expected_output_132

Problem 133: Week 3 Challenge 33 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_133

Output: expected_output_133

Problem 134: Week 3 Challenge 34 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_134

Output: expected_output_134

Problem 135: Week 3 Challenge 35 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_135

Output: expected_output_135

Problem 136: Week 3 Challenge 36 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_136

Output: expected_output_136

Problem 137: Week 3 Challenge 37 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_137

Output: expected_output_137

Problem 138: Week 3 Challenge 38 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_138

Output: expected_output_138

Problem 139: Week 3 Challenge 39 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_139

Output: expected_output_139

Problem 140: Week 3 Challenge 40 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_140

Output: expected_output_140

Problem 141: Week 3 Challenge 41 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_141

Output: expected_output_141

Problem 142: Week 3 Challenge 42 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_142

Output: expected_output_142

Problem 143: Week 3 Challenge 43 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_143

Output: expected_output_143

Problem 144: Week 3 Challenge 44 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_144

Output: expected_output_144

Problem 145: Week 3 Challenge 45 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_145

Output: expected_output_145

Problem 146: Week 3 Challenge 46 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_146

Output: expected_output_146

Problem 147: Week 3 Challenge 47 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_147

Output: expected_output_147

Problem 148: Week 3 Challenge 48 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_148

Output: expected_output_148

Problem 149: Week 3 Challenge 49 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_149

Output: expected_output_149

Problem 150: Week 3 Challenge 50 (Medium)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 3 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_150

Output: expected_output_150

Problem 151: Week 4 Challenge 1 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_151

Output: expected_output_151

Problem 152: Week 4 Challenge 2 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_152

Output: expected_output_152

Problem 153: Week 4 Challenge 3 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_153

Output: expected_output_153

Problem 154: Week 4 Challenge 4 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_154

Output: expected_output_154

Problem 155: Week 4 Challenge 5 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_155

Output: expected_output_155

Problem 156: Week 4 Challenge 6 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_156

Output: expected_output_156

Problem 157: Week 4 Challenge 7 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to

Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_157

Output: expected_output_157

Problem 158: Week 4 Challenge 8 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_158

Output: expected_output_158

Problem 159: Week 4 Challenge 9 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_159

Output: expected_output_159

Problem 160: Week 4 Challenge 10 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_160

Output: expected_output_160

Problem 161: Week 4 Challenge 11 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_161

Output: expected_output_161

Problem 162: Week 4 Challenge 12 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_162

Output: expected_output_162

Problem 163: Week 4 Challenge 13 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_163

Output: expected_output_163

Problem 164: Week 4 Challenge 14 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_164

Output: expected_output_164

Problem 165: Week 4 Challenge 15 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_165

Output: expected_output_165

Problem 166: Week 4 Challenge 16 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_166

Output: expected_output_166

Problem 167: Week 4 Challenge 17 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_167

Output: expected_output_167

Problem 168: Week 4 Challenge 18 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_168

Output: expected_output_168

Problem 169: Week 4 Challenge 19 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_169

Output: expected_output_169

Problem 170: Week 4 Challenge 20 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_170

Output: expected_output_170

Problem 171: Week 4 Challenge 21 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_171

Output: expected_output_171

Problem 172: Week 4 Challenge 22 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_172

Output: expected_output_172

Problem 173: Week 4 Challenge 23 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_173

Output: expected_output_173

Problem 174: Week 4 Challenge 24 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_174

Output: expected_output_174

Problem 175: Week 4 Challenge 25 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_175

Output: expected_output_175

Problem 176: Week 4 Challenge 26 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_176

Output: expected_output_176

Problem 177: Week 4 Challenge 27 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_177

Output: expected_output_177

Problem 178: Week 4 Challenge 28 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_178

Output: expected_output_178

Problem 179: Week 4 Challenge 29 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_179

Output: expected_output_179

Problem 180: Week 4 Challenge 30 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_180

Output: expected_output_180

Problem 181: Week 4 Challenge 31 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_181

Output: expected_output_181

Problem 182: Week 4 Challenge 32 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_182

Output: expected_output_182

Problem 183: Week 4 Challenge 33 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_183

Output: expected_output_183

Problem 184: Week 4 Challenge 34 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_184

Output: expected_output_184

Problem 185: Week 4 Challenge 35 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_185

Output: expected_output_185

Problem 186: Week 4 Challenge 36 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_186

Output: expected_output_186

Problem 187: Week 4 Challenge 37 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_187

Output: expected_output_187

Problem 188: Week 4 Challenge 38 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_188

Output: expected_output_188

Problem 189: Week 4 Challenge 39 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_189

Output: expected_output_189

Problem 190: Week 4 Challenge 40 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_190

Output: expected_output_190

Problem 191: Week 4 Challenge 41 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_191

Output: expected_output_191

Problem 192: Week 4 Challenge 42 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_192

Output: expected_output_192

Problem 193: Week 4 Challenge 43 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_193

Output: expected_output_193

Problem 194: Week 4 Challenge 44 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_194

Output: expected_output_194

Problem 195: Week 4 Challenge 45 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_195

Output: expected_output_195

Problem 196: Week 4 Challenge 46 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to

Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_196

Output: expected_output_196

Problem 197: Week 4 Challenge 47 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_197

Output: expected_output_197

Problem 198: Week 4 Challenge 48 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_198

Output: expected_output_198

Problem 199: Week 4 Challenge 49 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_199

Output: expected_output_199

Problem 200: Week 4 Challenge 50 (Hard)

Implement a GoLang function that solves a common problem in algorithms or data structures appropriate to Week 4 (e.g., arrays, strings, maps, concurrency, or file I/O).

Input: sample_input_200

Output: expected_output_200