Hackathon Description:

This hackathon is for sharpen your TypeScript skills with 100 challenging problems covering arrays, loops, and functions. Put your coding prowess to the test and showcase your problem-solving abilities!

Note:

1. Understand the problem thoroughly before writing code.
2. Use Google for technical information or formulas, not for direct answers.
3. Solve problems with your own thoughts and logic; avoid seeking immediate help.
4. Plan your solution before coding.
5. Test your code with various inputs and edge cases.
6. Seek help sparingly and ask specific questions.
7. Learn from others' solutions after attempting it yourself.
8. Stay persistent and patient when facing challenges.
9. Enjoy the process of problem-solving in coding.

Start Date: Saturday, 08-26-2023 at 12:00 AM
End Date: Sunday, 08-27-2023 at 12:00 PM

Arrays in TypeScript (30 problems):

1. Find the maximum number in an array.

Input: [3, 5, 1, 8, 10]
Output: 10

2. Calculate the sum of all numbers in an array.

Input: [2, 4, 6, 8]
Output: 20

3. Remove duplicates from an array.

Input: [1, 2, 2, 3, 4, 4, 5]
Output: [1, 2, 3, 4, 5]

4. Find the index of a specific element in an array.

Input: [10, 20, 30, 40, 50], 30
Output: 2

5. Sort an array of strings alphabetically.

Input: ["apple", "banana", "cherry", "date"]
Output: ["apple", "banana", "cherry", "date"]

6. Check if all elements in an array are even.

Input: [2, 4, 6, 8]
Output: true

7. Split an array into chunks of a specified size.

Input: [1, 2, 3, 4, 5, 6, 7, 8], 3
Output: [[1, 2, 3], [4, 5, 6], [7, 8]]

8. Find the intersection of two arrays.

Input: [1, 2, 3, 4], [3, 4, 5, 6]
Output: [3, 4]

9. Rotate an array to the right by a given number of positions.

Input: [1, 2, 3, 4, 5], 2
Output: [4, 5, 1, 2, 3]

10. Check if an array is a palindrome (reads the same forwards and backwards).
- Input: [1, 2, 3, 2, 1]
- Output: true

11. Merge two sorted arrays into one sorted array.
- Input: [1, 3, 5], [2, 4, 6]
- Output: [1, 2, 3, 4, 5, 6]

12. Find the difference between two arrays.
- Input: [1, 2, 3, 4], [2, 3]
- Output: [1, 4]

13. Count the occurrences of a specific element in an array.
- Input: [1, 2, 2, 3, 2, 4, 5], 2
- Output: 3

14. Find the second smallest number in an array.
- Input: [5, 2, 8, 1, 7]
- Output: 2

15. Find the longest consecutive sequence of numbers in an array.
- Input: [100, 4, 200, 1, 3, 2]
- Output: [1, 2, 3, 4]

16. Implement a stack using an array with push and pop methods.

17. Implement a queue using an array with enqueue and dequeue methods.

18. Flatten a nested array.
- Input: [1, [2], [3, [4]]]
- Output: [1, 2, 3, 4]

19. Calculate the product of all numbers in an array.
- Input: [2, 3, 4]
- Output: 24

20. Find the first non-repeating element in an array.
- Input: [1, 2, 3, 2, 1, 4]
- Output: 3

21. Remove the specified element from an array in-place.
- Input: [1, 2, 3, 4, 5], 3
- Output: [1, 2, 4, 5]

22. Check if two arrays are equal (have the same elements in the same order).
- Input: [1, 2, 3], [1, 2, 3]
- Output: true

23. Find the kth smallest element in an unsorted array.
- Input: [5, 3, 1, 2, 4], k=3
- Output: 3

24. Remove all falsy values (false, null, 0, "", undefined, and NaN) from an array.
- Input: [0, 1, false, true, null, undefined, NaN, "hello"]
- Output: [1, true, "hello"]

25. Reverse an array in-place.
- Input: [1, 2, 3, 4, 5]
- Output: [5, 4, 3, 2, 1]

26. Calculate the median of an array of numbers.
- Input: [4, 2, 7, 1, 5]
- Output: 4

27. Find the missing number in an array of consecutive numbers.
- Input: [1, 2, 4, 5, 6]
- Output: 3

28. Implement a binary search algorithm on a sorted array.

29. Remove elements from an array that satisfy a given condition.
- Input: [1, 2, 3, 4, 5], element => element % 2 === 0
- Output: [1, 3, 5]

30. Implement a function to shuffle the elements of an array randomly.

Loops (30 problems)

31. Print numbers from 1 to 10 using a for loop.
- Output: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

32. Print even numbers from 1 to 20 using a while loop.
- Output: 2, 4, 6, 8, 10, 12, 14, 16, 18, 20

33. Calculate the factorial of a number using a for loop.
- Input: 5
- Output: 120

34. Generate a Fibonacci sequence of n terms using a while loop.
- Input: 6
- Output: [0, 1, 1, 2, 3, 5]

35. Print the multiplication table for a given number using a for loop.
- Input: 3
- Output:
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
...
3 x 10 = 30

36. Calculate the sum of squares of numbers from 1 to n using a for loop.
- Input: 4
- Output: 30

37. Reverse a string using a for loop.
- Input: "hello"
- Output: "olleh"

38. Count the number of vowels in a string using a for loop.
- Input: "programming"
- Output: 3

39. Check if a given number is prime using a for loop.
- Input: 7
- Output: true

40. Print a pattern of stars using nested loops.
- Output:
* * * * * * * * * *

41. Calculate the sum of the first n natural numbers using a while loop.
- Input: 5
- Output: 15

42. Find the greatest common divisor (GCD) of two numbers using a loop.
- Input: 12, 18
- Output: 6

43. Implement a basic calculator that can perform addition, subtraction, multiplication, and division using a loop.

44. Calculate the power of a number using a loop.
- Input: 2, 3
- Output: 8

45. Check if a string is a palindrome using a loop.
- Input: "racecar"
- Output: true

46. Calculate the sum of all prime numbers up to a given number using a loop.
- Input: 10
- Output: 17

47. Implement a function to find the factorial of a number using recursion.

48. Implement a function to calculate the nth term of the Fibonacci sequence using recursion.

49. Implement a function to find the LCM (Least Common Multiple) of two numbers using recursion.

50. Implement a function to find the sum of digits of a number using recursion.

51. Implement a function to reverse a string using recursion.

52. Implement a function to calculate the power of a number using recursion.

53. Implement a function to check if a string is a palindrome using recursion.

54. Implement a function to solve the Tower of Hanoi puzzle using recursion.

55. Implement a function to generate all permutations of a string using recursion.

Functions for Loops, Arrays, and If-Else (40 problems)

56. Find the largest number in an array of numbers and return it from a function.

57. Calculate the average of numbers in an array and return it from a function.

58. Find and return the first even number in an array using a function.

59. Find and return the last odd number in an array using a function.

60. Check if a number is prime and return true or false from a function.

61. Calculate the sum of all multiples of 3 and 5 below a given number and return it from a function.

62. Find and return the index of the first occurrence of a specific element in an array using a function.

63. Calculate the factorial of a number using recursion and return it from a function.

64. Generate the nth term of the Fibonacci sequence using recursion and return it from a function.

65. Implement a function to reverse a string and return the reversed string.

66. Implement a function to check if a string is a palindrome and return true or false.

67. Find the smallest and largest numbers in an array and return them as an object from a function.

68. Implement a function to merge two sorted arrays into one sorted array and return it.

69. Calculate the product of all numbers in an array and return it from a function.

70. Implement a function to shuffle the elements of an array randomly and return the shuffled array.

71. Calculate the sum of all even numbers in an array and return it from a function.

72. Calculate the sum of all odd numbers in an array and return it from a function.

73. Implement a function to find the common elements between two arrays and return them in a new array.

74. Implement a function to remove all duplicates from an array and return the duplicated array.

75. Implement a function to rotate an array to the right by a given number of positions and return the rotated array.

76. Implement a function to find the intersection of two arrays and return it.

77. Implement a function to remove elements from an array that satisfy a given condition and return the modified array.

78. Implement a function that returns the first n prime numbers in an array.

79. Implement a function that returns the elements of an array in reverse order.

80. Implement a function that returns the index of the largest element in an array.

81. Implement a function to check if two arrays are equal (have the same elements in the same order) and return true or false.

82. Implement a function to find and return the kth smallest element in an array.

83. Implement a function that returns the difference between two arrays.

84. Implement a function that returns the longest consecutive sequence of numbers in an array.

85. Implement a function to find the missing number in an array of consecutive numbers.

86. Implement a function to find and return the first non-repeating element in an array.

87. Implement a function to check if all elements in an array are even and return true or false.

88. Implement a function to split an array into chunks of a specified size and return an array of chunks.

89. Implement a function to check if a given array is a palindrome (reads the same forwards and backwards) and return true or false.

90. Implement a function to remove all falsy values (false, null, 0, "", undefined, NaN) from an array and return the modified array.

91. Implement a function to reverse an array in-place.

92. Implement a function to calculate the median of an array of numbers.

93. Implement a function to find the second smallest number in an array.

94. Implement a function to calculate the sum of squares of numbers in an array.

95. Implement a function to flatten a nested array and return a flat array.

96. Implement a function to implement a stack using an array with push and pop methods.

97. Implement a function to implement a queue using an array with enqueue and dequeue methods.

98. Implement a function to find the greatest common divisor (GCD) of two numbers using a loop and return it.

99. Implement a function to print a pattern of stars using nested loops.

100. Implement a function to calculate the LCM (Least Common Multiple) of two numbers using recursion and return it.