## Easy Phase (1-30)

1. Use `map` to create a new array that doubles the values of the given array `[1, 2, 3, 4]`.
2. Use `filter` to remove even numbers from the array `[5, 8, 12, 15, 18]`.
3. Use `reduce` to sum up all elements in the array `[10, 20, 30, 40]`.
4. Use `find` to get the first element greater than 15 in the array `[8, 12, 16, 20, 24]`.
5. Use `findIndex` to locate the index of the first odd number in the array `[2, 4, 6, 9, 12]`.
6. Use `some` to check if there is any number greater than 100 in `[10, 20, 30, 150]`.
7. Use `every` to check if all numbers in the array `[5, 10, 15, 20]` are multiples of 5.
8. Use `sort` to sort the array `[3, 5, 1, 4, 2]` in ascending order.
9. Use `filter` to extract words longer than 3 characters from `["cat", "dog", "fish", "elephant"]`.
10. Use `map` to convert an array of prices `[10, 20, 30]` to include tax (10%).
11. Use `reduce` to multiply all numbers in the array `[2, 3, 4]`.
12. Use `sort` to arrange `["banana", "apple", "cherry"]` alphabetically.
13. Use `some` to check if any element in `[12, 24, 35, 47]` is divisible by 5.
14. Use `find` to get the first negative number in `[7, -3, 9, -8, 2]`.
15. Use `every` to check if all numbers in `[9, 18, 27, 36]` are divisible by 9.
16. Use `filter` to get names starting with "A" from `["Alice", "Bob", "Amanda", "Carl"]`.
17. Use `map` to square all elements of `[1, 2, 3, 4, 5]`.
18. Use `find` to get the first string longer than 4 characters from `["dog", "horse", "cat", "sheep"]`.
19. Use `sort` to sort the numbers `[50, 20, 10, 40, 30]` in descending order.
20. Use `reduce` to concatenate an array of strings `["Hello", " ", "World"]`.
21. Use `map` to transform `["John", "Doe"]` into `["john", "doe"]` (lowercase).
22. Use `filter` to get all elements less than 100 from `[150, 80, 90, 300]`.
23. Use `findIndex` to locate the position of "cat" in `["dog", "cat", "bird"]`.
24. Use `every` to confirm if all elements in `[10, 20, 30, 40]` are greater than 5.
25. Use `some` to determine if any number in `[25, 35, 45]` is greater than 40.
26. Use `filter` to extract odd numbers from `[12, 15, 19, 21, 24]`.
27. Use `map` to append the word "Item" to each element of `["One", "Two", "Three"]`.
28. Use `sort` to sort an array of characters `["b", "a", "c", "e", "d"]` alphabetically.
29. Use `reduce` to find the maximum number in `[3, 6, 2, 8, 5]`.

30. Use `find` to get the first number greater than 100 in `[50, 120, 90, 150]`.

---

## Medium Phase (31-60)

31. Use `reduce` to flatten an array `[[1, 2], [3, 4], [5, 6]]` into a single array.
32. Use `filter` to remove duplicates from `[1, 2, 3, 1, 2, 4, 5]`.
33. Use `map` to generate an array of lengths of each word in `["apple", "banana", "cherry"]`.
34. Use `findIndex` to locate the index of the first number divisible by 7 in `[14, 28, 35, 21]`.
35. Use `some` to check if the array `["apple", "banana", "grape"]` contains the word "banana".
36. Use `reduce` to count the occurrences of each character in the string "hello world".
37. Use `sort` to sort an array of objects by age `[{age: 30}, {age: 25}, {age: 35}]`.
38. Use `filter` to get people older than 25 from `[{age: 20}, {age: 30}, {age: 40}]`.
39. Use `map` to create a new array of full names from `[{first: "John", last: "Doe"}]`.
40. Use `reduce` to calculate the total price from an array of objects `[{price: 10}, {price: 20}]`.
41. Use `find` to get the object where the `id` is 3 from `[{id: 1}, {id: 3}, {id: 5}]`.
42. Use `filter` to get all the even numbers squared from `[1, 2, 3, 4, 5]`.
43. Use `sort` to arrange strings by length `["short", "longer", "longest"]`.
44. Use `reduce` to implement a custom `map` function.
45. Use `every` to check if all objects in `[{age: 25}, {age: 30}]` are above 20.
46. Use `map` to extract the domains from emails `["user1@domain.com", "user2@site.net"]`.
47. Use `findIndex` to get the index of the first number greater than 10 in `[2, 8, 11, 15]`.
48. Use `reduce` to reverse a string "javascript".
49. Use `filter` to keep only positive numbers from `[-10, 20, -30, 40, 50]`.
50. Use `some` to determine if the array `[NaN, 5, 10]` contains `NaN`.
51. Use `map` to remove vowels from an array of strings `["apple", "banana"]`.
52. Use `sort` to sort an array of dates `["2024-10-01", "2022-03-01"]` chronologically.
53. Use `reduce` to implement a basic factorial calculation.
54. Use `filter` to extract words containing only vowels from `["eye", "sky", "tooth"]`.
55. Use `find` to get the first palindrome in `["car", "madam", "racecar"]`.

56. Use `map` to transform each number in `[1, 2, 3]` to its English word equivalent.
57. Use `reduce` to implement a custom `filter` function.
58. Use `sort` to arrange an array of objects by the length of their `name` property.
59. Use `some` to check if any string in `["123", "abc", "45"]` contains only numbers.
60. Use `every` to validate that all elements in `[{id: 1}, {id: 2}]` have an `id` property.

---

## Hard Phase (61-80)

61. Use `reduce` to implement a deep object merging for `[{a: 1}, {b: 2}, {a: 3}]`.
62. Use `filter` to get all prime numbers from an array `[1, 2, 3, 4, 5, 6, 7, 8, 9]`.
63. Use `map` to apply a function that returns the factorial of each number in `[1, 2, 3, 4]`.
64. Use `find` to get the first object where the `name` starts with "J" from `[{name: "John"}, {name: "Alex"}]`.
65. Use `sort` to sort an array of objects by multiple keys `[{age: 25, score: 90}]`.
66. Use `reduce` to build a frequency map of words from `["apple", "banana", "apple"]`.
67. Use `map` to create an array of URLs from `["google", "github"]`.
68. Use `findIndex` to locate the first object with a `value` property less than 5 in `[{}]`.
69. Use `filter` to create a new array without nested arrays from `[[1, 2], 3, [4, 5], 6]`.
70. Use `reduce` to implement a custom `find` function.
71. Use `sort` to organize words by the number of vowels in each word `["apple", "orange"]`.
72. Use `every` to validate an array of emails `["test@domain.com", "admin@site.org"]`.
73. Use `reduce` to implement a method that compresses an array to unique values.
74. Use `map` to transform an array of arrays `[[1, 2], [3, 4]]` to a single-level array.
75. Use `filter` to get people with unique IDs from an array of objects with repeated IDs.
76. Use `reduce` to find the most common character in a string "programming".
77. Use `find` to locate the object with the maximum `age` property.
78. Use `some` to detect if an array contains a nested array.
79. Use `map` to apply a series of functions to each number in `[10, 20, 30]`.
80. Use `sort` to order objects by a computed property.