

EASY QUESTIONS:

1. Reverse a String: Reverse the given string. For example, "hello" becomes "olleh".
2. Check Palindrome: Check if the given string is a palindrome, meaning it reads the same forwards and backwards.
3. Fibonacci Series: Generate the Fibonacci series up to a given number or term. Each term in the series is the sum of the two preceding ones, starting from 0 and 1.
4. Find Maximum Element in Array: Find the maximum element in a given array of integers.
5. Check Anagram: Check if two given strings are anagrams of each other, meaning they contain the same characters in a different order.
6. Check Prime Number: Check if a given number is a prime number, meaning it is only divisible by 1 and itself.
7. Find Factorial: Compute the factorial of a given non-negative integer.
8. Count Vowels and Consonants: Count the number of vowels and consonants in a given string.
9. Check Armstrong Number: Check if a given number is an Armstrong number, meaning the sum of its own digits raised to the power of the number of digits is equal to the number itself.
10. Calculate Simple Interest: Calculate the simple interest given principal amount, rate, and time.
11. Calculate Compound Interest: Calculate the compound interest given principal amount, rate, and time.
12. Find GCD and LCM: Find the greatest common divisor (GCD) and least common multiple (LCM) of two given numbers.
13. Check Leap Year: Check if a given year is a leap year.
14. Convert Decimal to Binary: Convert a decimal number to its binary equivalent.
15. Convert Decimal to Hexadecimal: Convert a decimal number to its hexadecimal equivalent.
16. Convert Decimal to Octal: Convert a decimal number to its octal equivalent.
17. Convert Binary to Decimal: Convert a binary number to its decimal equivalent.
18. Convert Hexadecimal to Decimal: Convert a hexadecimal number to its decimal equivalent.

19. Convert Octal to Decimal: Convert an octal number to its decimal equivalent.
20. Reverse a Number: Reverse the digits of a given integer.
21. Sum of Digits: Find the sum of the digits of a given integer.
22. Product of Digits: Find the product of the digits of a given integer.
23. Check Perfect Number: Check if a given number is a perfect number, meaning the sum of its proper divisors equals the number itself.
24. Find Factorial of Large Numbers: Compute the factorial of large numbers that may not fit in standard data types.
25. Check Even or Odd: Check if a given number is even or odd.
26. Find Power of a Number: Compute the power of a given number raised to a specified exponent.
27. Find Roots of Quadratic Equation: Find the roots of a quadratic equation given its coefficients.
28. Reverse Words in a String: Reverse the words in a given string while maintaining the order of words.
29. Check Pangram Sentence: Check if a given sentence is a pangram, meaning it contains every letter of the alphabet at least once.
30. Check Armstrong Number in a Range: Check for Armstrong numbers in a given range of numbers.
31. Check Prime Number in a Range: Check for prime numbers in a given range of numbers.
32. Find Sum of Natural Numbers: Find the sum of the first n natural numbers.
33. Find Sum of Squares of Natural Numbers: Find the sum of squares of the first n natural numbers.
34. Find Sum of Cubes of Natural Numbers: Find the sum of cubes of the first n natural numbers.
35. Check Strong Number: Check if a given number is a strong number, meaning the sum of the factorial of its digits equals the number itself.
36. Check Harshad Number: Check if a given number is a Harshad number, meaning it is divisible by the sum of its digits.
37. Check Disarium Number: Check if a given number is a Disarium number, meaning the sum of its digits raised to their respective positions equals the number itself.
38. Check Neon Number: Check if a given number is a Neon number, meaning the sum of the digits of its square equals the number itself.

39. Check Abundant Number: Check if a given number is an abundant number, meaning the sum of its proper divisors is greater than the number itself.
40. Check Deficient Number: Check if a given number is a deficient number, meaning the sum of its proper divisors is less than the number itself.
41. Check Happy Number: Check if a given number is a happy number, meaning the sequence of summing the squares of its digits eventually reaches 1.
42. Check Magic Number: Check if a given number is a magic number, meaning the repeated sum of its digits eventually reaches 1.
43. Find Nth Fibonacci Number: Find the nth Fibonacci number in the Fibonacci sequence.
44. Check if a Number is a Power of Two: Check if a given number is a power of two.
45. Find Largest Among Three Numbers: Find the largest among three given numbers.
46. Find Smallest Among Three Numbers: Find the smallest among three given numbers.
47. Find Factors of a Number: Find all the factors of a given number.
48. Find Sum of Digits of a Number: Find the sum of the digits of a given number.
49. Check Smith Number: Check if a given number is a Smith number, meaning the sum of the digits of its prime factors equals the sum of its digits.
50. Calculate Natural Logarithm: Calculate the natural logarithm of a given number.
51. Calculate Logarithm to the Base 10: Calculate the logarithm of a given number to the base 10.
52. Calculate Exponential Value: Calculate the exponential value of a given number.
53. Check Automorphic Number: Check if a given number is an automorphic number, meaning its square ends with the same digits as the number itself.
54. Find Average of Numbers in an Array: Compute the average of all the numbers in a given array.
55. Check Prime Numbers in a Range: Checking prime numbers in a range involves identifying all prime numbers within a specified range of integers. A prime number is a natural number greater than 1 that has no positive divisors other than 1 and itself.
56. Find Prime Factors of a Number: Finding prime factors of a number involves identifying all the prime numbers that divide the given number exactly without leaving a remainder.
57. Generate Random Numbers: Generating random numbers involves producing numbers that have no discernible pattern and are uniformly distributed over a specified range.

58. Generate Random Password: Generating a random password involves creating a sequence of characters, typically including a mix of letters, numbers, and special symbols, for use as a secure login credential.
59. Find Quadrant of a Coordinate: Finding the quadrant of a coordinate involves determining in which of the four quadrants (I, II, III, or IV) of the Cartesian coordinate system a point lies based on its x and y coordinates.
60. Convert Temperature Units: Converting temperature units involves converting temperatures from one unit of measurement (e.g., Celsius, Fahrenheit, Kelvin) to another.
61. Convert Units of Length: Converting units of length involves converting measurements from one unit of length (e.g., meters, centimeters, inches, feet) to another.
62. Convert Units of Weight: Converting units of weight involves converting measurements from one unit of weight (e.g., grams, kilograms, ounces, pounds) to another.
63. Convert Units of Area: Converting units of area involves converting measurements from one unit of area (e.g., square meters, square feet, square kilometers) to another.
64. Convert Units of Volume: Converting units of volume involves converting measurements from one unit of volume (e.g., cubic meters, cubic feet, liters, gallons) to another.
65. Find Perimeter of Rectangle: The perimeter of a rectangle is the total length of its four sides. It can be calculated using the formula: $\text{perimeter} = 2 * (\text{length} + \text{width})$.
66. Find Perimeter of Circle: The perimeter of a circle, also known as the circumference, is the distance around its outer edge. It can be calculated using the formula: $\text{perimeter} = 2 * \pi * \text{radius}$.
67. Find Area of Rectangle: The area of a rectangle is the measure of the region enclosed by its sides. It can be calculated using the formula: $\text{area} = \text{length} * \text{width}$.
68. Find Area of Circle: The area of a circle is the measure of the region enclosed by its circumference. It can be calculated using the formula: $\text{area} = \pi * \text{radius}^2$.
69. Find Area of Triangle: The area of a triangle is the measure of the region enclosed by its three sides. It can be calculated using various formulas, such as Heron's formula or by using the base and height of the triangle.
70. Find Volume of Sphere: The volume of a sphere is the measure of the space enclosed by its surface. It can be calculated using the formula: $\text{volume} = (4/3) * \pi * \text{radius}^3$.
71. Find Volume of Cylinder: The volume of a cylinder is the measure of the space enclosed by its curved surface. It can be calculated using the formula: $\text{volume} = \pi * \text{radius}^2 * \text{height}$.

72. Find Volume of Cone: The volume of a cone is the measure of the space enclosed by its surface. It can be calculated using the formula: $\text{volume} = (1/3) * \pi * \text{radius}^2 * \text{height}$.
73. Find Volume of Cube: The volume of a cube is the measure of the space enclosed by its six square faces. It can be calculated using the formula: $\text{volume} = \text{side}^3$.
74. Find Volume of Cuboid: The volume of a cuboid is the measure of the space enclosed by its six rectangular faces. It can be calculated using the formula: $\text{volume} = \text{length} * \text{width} * \text{height}$.
75. Find Volume of Prism: The volume of a prism is the measure of the space enclosed by its surface. It can be calculated by multiplying the area of its base by its height.
76. Find Volume of Pyramid: The volume of a pyramid is the measure of the space enclosed by its surface. It can be calculated using the formula: $\text{volume} = (1/3) * \text{base area} * \text{height}$.
77. Find Perimeter of Triangle: The perimeter of a triangle is the total length of its three sides. It can be calculated by summing the lengths of the three sides.
78. Find Perimeter of Square: The perimeter of a square is the total length of its four equal sides. It can be calculated using the formula: $\text{perimeter} = 4 * \text{side length}$.
79. Find Perimeter of Rhombus: The perimeter of a rhombus is the total length of its four equal sides. It can be calculated by multiplying the length of one side by 4.
80. Find Perimeter of Parallelogram: The perimeter of a parallelogram is the total length of its four sides. It can be calculated by summing the lengths of the opposite sides.
81. Find Perimeter of Trapezium: The perimeter of a trapezium (or trapezoid) is the total length of its four sides. It can be calculated by summing the lengths of all its sides.
82. Find Circumference of Circle: The circumference of a circle is the distance around its outer edge. It can be calculated using the formula: $\text{circumference} = 2 * \pi * \text{radius}$.
83. Find Area of Square: The area of a square is the measure of the region enclosed by its four equal sides. It can be calculated using the formula: $\text{area} = \text{side length}^2$.
84. Find Area of Rhombus: The area of a rhombus is the measure of the region enclosed by its four equal sides. It can be calculated using the formula: $\text{area} = (\text{diagonal1} * \text{diagonal2}) / 2$.
85. Find Area of Parallelogram: The area of a parallelogram is the measure of the region enclosed by its four sides. It can be calculated using the formula: $\text{area} = \text{base} * \text{height}$.
86. Find Area of Trapezium: The area of a trapezium (or trapezoid) is the measure of the region enclosed by its four sides. It can be calculated using the formula: $\text{area} = (1/2) * (\text{sum of parallel sides}) * \text{height}$.

87. Find Area of Regular Polygon: The area of a regular polygon is the measure of the region enclosed by its sides, where all sides and angles are equal. It can be calculated using various formulas depending on the specific polygon.
88. Find Area of Sector: The area of a sector of a circle is the measure of the region enclosed by an arc and two radii. It can be calculated using the formula: $\text{area} = (\theta/360) * \pi * \text{radius}^2$, where θ is the central angle of the sector in degrees.
89. Find Area of Segment: The area of a segment of a circle is the measure of the region enclosed by an arc and a chord. It can be calculated using various methods depending on the specific segment shape.
90. Find Area of Ellipse: The area of an ellipse is the measure of the region enclosed by its perimeter. It can be calculated using the formula: $\text{area} = \pi * \text{major axis} * \text{minor axis}$.
91. Floor in a Sorted Array: Given a sorted array `arr[]` of size `N` without duplicates, and given a value `x`. Floor of `x` is defined as the largest element `K` in `arr[]` such that `K` is smaller than or equal to `x`. Find the index of `K` (0-based indexing)
92. Intersection of two arrays: Given two arrays `a[]` and `b[]` respectively of size `n` and `m`, the task is to print the count of elements in the intersection (or common elements) of the two arrays.
93. Queue using two Stacks: Implement a Queue using 2 stacks `s1` and `s2` .
A Query `Q` is of 2 Types
(i) 1 `x` (a query of this type means pushing '`x`' into the queue)
(ii) 2 (a query of this type means to pop element from queue and print the popped element)
94. Convert array into Zig-Zag fashion: Given an array `arr` of distinct elements of size `N`, the task is to rearrange the elements of the array in a zig-zag fashion so that the converted array should be in the below form:
- `arr[0] < arr[1] > arr[2] < arr[3] > arr[4] < . . . arr[n-2] < arr[n-1] > arr[n]`.
95. Maximize Toys: Given an array `arr[]` of length `N` consisting cost of `N` toys and an integer `K` depicting the amount with you. Your task is to find maximum number of toys you can buy with `K` amount.
96. Plus One: Given a non-negative number represented as a list of digits, add 1 to the number (increment the number represented by the digits). The digits are stored such that the most significant digit is first element of array.
97. Save Ironman: Jarvis is weak in computing palindromes for Alphanumeric characters. While Ironman is busy fighting Thanos, he needs to activate sonic punch but Jarvis is stuck in computing palindromes.

You are given a string S. Find out whether the alphanumeric characters of the string form a palindrome or not.

If you are unable to solve it then it may result in the death of Iron Man.

98. Intersection of Two Linked Lists: Given two linked lists, the task is to complete the function `findIntersection()`, that returns the intersection of two linked lists. Each of the two linked list contains distinct node values.
99. Product Pair: Given an array `arr[]` of size N of distinct elements and a number X, find if there is a pair in `arr[]` with product equal to X.
100. Merge Two Sorted Arrays: Given two sorted arrays `arr1` and `arr2` of sizes m and n respectively, merge them into a single sorted array without using extra space.
101. First Unique Character in a String: Given a string, find the first non-repeating character in it and return its index. If it doesn't exist, return -1.
102. Minimum Size Subarray Sum: Given an array of positive integers `nums` and a positive integer `target`, return the minimal length of a contiguous subarray [`numsl`, `numsl+1`, ..., `numsr-1`, `numsr`] of which the sum is greater than or equal to `target`.
103. Maximum Length of Pair Chain: You are given an array of n pairs `pairs` where `pairs[i] = [a, b]` and `pairs[i][0] < pairs[i][1]`. A pair chain is an ordered set of pairs (a, b) where `pairs[i][0] < pairs[i+1][0]`. Return the length of the longest possible pair chain.
104. Find Peak Element: A peak element in an array is an element that is strictly greater than its neighbors. Given an integer array `nums`, find a peak element, and return its index. If the array contains multiple peaks, return the index to any of the peaks.
105. Longest Consecutive Sequence: Given an unsorted array of integers `nums`, return the length of the longest consecutive elements sequence.
106. Container With Most Water: Given n non-negative integers `a1`, `a2`, ..., `an`, where each represents a point at coordinate (i, `ai`). n vertical lines are drawn such that the two endpoints of the line i is at (i, `ai`) and (i, 0). Find two lines, which, together with the x-axis forms a container, such that the container contains the most water.
107. 3Sum: Given an array `nums` of n integers, are there elements a, b, c in `nums` such that $a + b + c = 0$? Find all unique triplets in the array which gives the sum of zero.
108. 4Sum: Given an array `nums` of n integers, return an array of all the unique quadruplets [`nums[a]`, `nums[b]`, `nums[c]`, `nums[d]`] such that: $- 0 \leq a, b, c, d < n$ - a, b, c, and d are distinct. - $nums[a] + nums[b] + nums[c] + nums[d] == target$
109. Group Anagrams: Given an array of strings `strs`, group the anagrams together. You can return the answer in any order.

110. Combination Sum: Given an array of distinct integers candidates and a target integer target, return a list of all unique combinations of candidates where the chosen numbers sum to target. You may return the combinations in any order.
111. Combination Sum II: Given a collection of candidate numbers (candidates) and a target number (target), find all unique combinations in candidates where the candidate numbers sum to target. Each number in candidates may only be used once in the combination.
112. Combination Sum III: Find all valid combinations of k numbers that sum up to n such that the following conditions are true: - Only numbers 1 through 9 are used. - Each number is used at most once.
113. Combination Sum IV: Given an array of distinct integers nums and a target integer target, return the number of possible combinations that add up to target.
114. Word Search: Given an m x n grid of characters board and a string word, return true if word exists in the grid. The word can be constructed from letters of sequentially adjacent cells, where adjacent cells are horizontally or vertically neighboring. The same letter cell may not be used more than once.
115. Permutations: Given an array nums of distinct integers, return all the possible permutations. You can return the answer in any order.
116. Permutations II: Given a collection of numbers, nums, that might contain duplicates, return all possible unique permutations in any order.
117. Rotate Image: You are given an n x n 2D matrix representing an image, rotate the image by 90 degrees (clockwise).
118. Reverse Linked List: Reverse a singly linked list.
119. Linked List Cycle: Given head, the head of a linked list, determine if the linked list has a cycle in it. There is a cycle in a linked list if there is some node in the list that can be reached again by continuously following the next pointer. Internally, pos is used to denote the index of the node that tail's next pointer is connected to. Note that pos is not passed as a parameter.
120. Linked List Cycle II: Given a linked list, return the node where the cycle begins. If there is no cycle, return null.