Easy-Level Questions

1.Write a function to check the grade of a student based on the score:

- 90-100: A
- 80-89: B
- 70-79: C
- 60-69: D
- Below 60: F

2. Write a function to determine whether a given number is positive, negative, or zero.

3. Write a function that takes three sides of a triangle and checks whether the triangle is equilateral, isosceles, or scalene.

4. Write a function to check whether a given character is a vowel or consonant.

5. Write a function that checks if a person is eligible to vote. The person is eligible if they are 18 years old or older.

6. Write a function that checks if a username and password match predefined values. If the username is "admin" and the password is "1234", print "Login successful"; otherwise, print "Login failed."

7. Write a function that simulates a traffic light system. The function should take the current light color (red, yellow, green) as input and print the corresponding action:

- "red" → "Stop"
- "yellow" → "Slow down"
- "green" → "Go"

8. Find the Average of an Array.

9. Sort an Array in Ascending Order **Example**: Input: `nums = [4,2,8,5,1];` Output: `nums = [1,2,4,5,8].`

10. `Given an array of integers, count how many numbers are even and how many are odd.`

Example

- **Input:** `[1, 2, 3, 4, 5]`
- **Output:** `Even: 2, Odd: 3`

11. Remove duplicate elements from the array `arr = [1, 2, 2, 3, 4, 4, 5]` and print the updated array.

12. Add the number `6` to the end of the array `arr = [1, 2, 3, 4, 5]` and print the updated array.

13. Check if the array `arr = [1, 2, 3, 4, 5]` contains the number `3` and print `true` or `false`.

14. Add Element to the Beginning of an Array **Example**: `Input: nums = [1, 2, 3, 4]; Output: nums = [0,1,2,3,4].`

15. Remove the Last element `Input: nums = [1, 2, 3, 4,5]; Output: nums = [1,2,3,4].`

16. Check if all the elements in `arr = [3, 5, 9, 1, 7]` are positive numbers, and print `true` or `false`.

17. Count how many positive and negative numbers are in `arr = [1, -2, 3, -4, 5, -6]` and print the result.

18. Print all elements that are at even indexes in the array `arr = [10, 20, 30, 40, 50]`.

19. Check if the array `arr = [1, 2, 3, 4, 5]` is sorted in ascending order, and print `true` or `false`.

20. Find and print the difference between the maximum and minimum elements in `arr = [80, 30, 70, 50, 20]`.

21. Write a program to convert a given string to uppercase.**Example**: Input: `"hello"`, Output: `"HELLO"`.

22. Write a program to find the length of a given string.

23. Write a program to concatenate two given strings. **Example**: Input: ("hello", " world"), Output: "hello world".

24. Write a program to remove whitespace from both ends of a string.**Example**: Input: " hello ", Output: "hello".

25. Write a program to split a string into an array of words.

26. Write a program to check if a string ends with a specific character.**Example**: Input: ("codinggita", "a"), Output: true.

27. Write a program to extract the file extension from a given filename.**Example**: Input: "document.pdf", Output: "pdf".

28. Write a function that takes two numbers and prints the largest one.

29.Write a program to find all pairs in an array whose sum is equal to a given number.

Example 1: **Input:** nums = [2,7,11,15], target = 9 , **Output:** [0,1].

30. Write a program to input an integer ***'n'*** and print the sum of all its even digits and the sum of all its odd digits separately. **Example : Input:** 'n' = 132456, **Output:** 12, 9

Explanation:

The sum of even digits = 2 + 4 + 6 = 12

The sum of odd digits = 1 + 3 + 5 = 9

31. Write a program to repeat a string a specified number of times.**Example**: **Input:** `("hello", 3)`, **Output:** `"hellohellohello".`

32.Write a program that categorizes a person's age group based on the given age:

- Less than 13: "Child"
- Between 13 and 19: "Teenager"
- Between 20 and 59: "Adult"
- 60 and above: "Senior"

33. Write a program that takes a year as input and checks whether it is a century year (a year divisible by 100).

34. Access and print the first and last element of the array arr = [10, 20, 30, 40, 50].

35. Print an inverted right-angled triangle pattern with n rows.

36. Print a pyramid pattern with n rows.

37. Given a sorted array and a target value, return the starting and ending position of that target in the array.**Example**:**Input:** [5, 7, 7, 8, 8, 10], **target=**8 ,**Output:** [3, 4]

38. Given a temperature in Celsius, convert it to Fahrenheit. **Example:Input:** 0 **Output:** 32.

39. Given a string, check if all brackets are closed properly. **Example:Input:** "{[()]}" **Output:** true

40. Given two numbers, generate an array containing all numbers between them

(inclusive). **Example: Input:** `1,5` **Output:** `[1, 2, 3, 4, 5]`

41. Given a valid IP address, you are asked to return a defanged version of that IP address. A defanged IP address replaces every period `"."` with `"[.]"`.

Example1:Input: address = "1.1.1.1" **Output:** "1[.]1[.]1[.]1"

Example2:Input: address = "255.100.50.0" **output:** "255[.]100[.]50[.]0"

42. Given two lists of events. Each event is represented by a start time and an end time. You need to determine if the two events conflict, which means if the events overlap in time.

Input Format:

- Each event is represented by a list `[start, end]`, where `start` is the start time (inclusive) and `end` is the end time (exclusive).
- The events are represented as two arrays: `event1` and `event2`.

Output:

- Return `true` if there is a conflict between the two events; otherwise, return `false`.

Example 1: Input: **event1 =** [1, 5], **event2 =** [5, 10] Output: false, **Exampe 2:** Input: **event1 =** [1, 5], **event2 =** [2, 3] Output: true.

43. The "Max Consecutive Ones" problem is a common algorithmic challenge that involves finding the maximum number of consecutive 1s in a binary array.

Problem Statement

50. Given a binary array, find the maximum number of consecutive 1s in the array.

Example

- **Input:** `[1, 1, 0, 1, 1, 1]`
- **Output:** 3 (the longest sequence of 1s is 111)

44. Given a string, return all possible substrings of that string. This includes all substrings of every length, from length 1 to the length of the string itself.

Example:

- **Input:** `"abc"`
- **Output:** `["a", "ab", "abc", "b", "bc","c"]`

45. Given a sentence, return the longest word in it.

- **Input:** `"I love programming in JavaScript"`
- **Output:** `"programming"`

46. Given a string, return the index of the first repeating character. If no character repeats, return -1.

- **Input:** "hello"
- **Output:** 2 (because 'l' repeats first)
- **Input:** "abcdef"
- **Output:** -1

47. Given an array of integers, find the first element that repeats. If no element repeats, return -1.

- **Input:** [10, 5, 3, 4, 3, 5, 6]
- **Output:** 5

48.Given a string, return a new string with all vowels removed.

- **Input:** "hello"
- **Output:** "hll"

49. Given an array and two indices, swap the elements at those indices.

- **Input:** arr = [1, 2, 3, 4], i = 1, j = 3
- **Output:** [1, 4, 3, 2]

50. Given a string and a character, count how many times the character appears in the string.

- **Input:** `str = "hello world", char = "o"`
- **Output:** `2`

51. Given two arrays, one containing keys and the other containing values, create an object that combines them.

- **Input:**
  - `keys = ['name', 'age', 'city']`
  - `values = ['Alice', 30, 'New York']`

Output:`{name: 'Alice', age: 30, city: 'New York'}`

52.You are given an array `stones[]` where each element represents a number of stones in a heap. You and your friend take turns to remove 1 to 3 stones from one of the heaps. The player who removes the last stone from the last heap wins. Determine if you can win the game assuming both you and your friend play optimally.**Example1:**

Input:stones=[4,1,6]**Output**:false **Example2:**
Input:stones =[5,3,2]**Output:** true.

Moderate-Level Questions

1. Write a function that takes a digit (0-9) as input and returns the corresponding word. For example, input `1` should return `"one"`.

2. Write a function that takes an hour (0-23) and prints an appropriate greeting based on the time:

- 5:00-11:59 → "Good morning"
- 12:00-17:59 → "Good afternoon"
- 18:00-21:59 → "Good evening"
- 22:00-4:59 → "Good night"

3. Write a simple calculator function that takes two numbers and an operator (`+`, `-`, `*`, `/`) as input, then returns the result of the operation.

4. Write a function that simulates a simple login system. The user has 3 attempts to input the correct password. After 3 failed attempts, the function should print "Account locked."

5. Write a function that takes three numbers as input and returns the second largest number.

6. Write a program that prints numbers from 1 to 100. But for multiples of 3, print `"Fizz"` instead of the number, and for multiples of 5, print `"Buzz"`. For numbers that are multiples of both 3 and 5, print `"FizzBuzz"`.

7. Rotate the array `arr = [1, 2, 3, 4, 5]` to the left by one position and print the updated array **Output:** [2, 3, 4, 5, 1].

8. Given an array `arr = [1, 2, 3, 4, 5, 6, 7, 8, 9]` and a target sum `sum = 10`, find all pairs of numbers that add up to the sum.

9. Rotate the array `arr = [1, 2, 3, 4, 5]` by `k = 2` positions to the right .**Output:** [4, 5, 1, 2, 3]

10. Merge two sorted arrays `arr1 = [1, 3, 5]` and `arr2 = [2, 4, 6]` into one sorted array and print the result.

11. Given an array `arr = [1, -2, 3, -4, 5, -6]`, rearrange it so that positive and negative numbers alternate.

12. Write a program to find the maximum element in an array. **Example**: Input: `[1, 3, 5, 2, 4]`, Output: `5`.

13. Write a program to count the number of vowels in a string.**Example**: Input: "hello", Output: 2.

14. Find and print the index of the number 4 in the array arr = [10, 20, 30, 40, 50].

15. Remove duplicate elements from the array arr = [1, 2, 2, 3, 4, 4, 5] and print the updated array.

16. Find and print the second largest element in the array arr = [10, 20, 5, 30, 15].

17. Rotate the array arr = [1, 2, 3, 4, 5] to the right by one position and print the updated array **Output:** [5, 1, 2, 3, 4].

18. Move all the zeroes in the array arr = [1, 0, 2, 0, 3, 0, 4] to the end.

19. Check if two strings are anagrams of each other. **Example**: Input:let str1 = "listen"; let str2 = "silent"; output:true.

20. You are given a non-negative integer `n`. Your task is to calculate the difference between the product of its digits and the sum of its digits.

Specifically, you need to:

- Find the product of the digits of `n`.
- Find the sum of the digits of `n`.
- Return the difference: **(Product of digits) - (Sum of digits)**.

21. Write a program to create a new array where each element is the square of the corresponding element in the original array
Example: Input: `nums = [1, 2, 3, 4];`
Output: `nums = [1,4,9,16].`

23.You are given two strings, `word1` and `word2`. Merge the strings by adding letters in alternating order, starting with `word1`. If one string is longer than the other, append the remaining letters from the longer string to the end of the merged string. Return the merged string.

.**Example**:Input:**word1=**"abc",**word2=**"pqr", Output:"apbqcr".

24.Given two strings s and t, return true *if* s *is a* ***subsequence*** *of* t*, or* false *otherwise*. **Example 1:Input: s=**"abc", **t=**"ahbgdc" **output:**true, **Example 2: input:** s="axc", **t=**"ahbgdc" **Output:** false.

25. Replace all negative numbers in `arr = [-1, 2, -3, 4, -5]` with zero and print the updated array. **Output:** [0, 2, 0, 4, 0].

26. Given an array where each element represents the stock price on a given day, find the maximum profit that can be made by buying and then selling the stock. **Example**: `arr = [7, 1, 5, 3, 6, 4]`, output should be 5 (buy on day 2 and sell on day 5).

27. Write a program that takes three numbers as input and prints the largest of the three using `if-else` statements.

28. Write a program to calculate the electricity bill based on the following rates:

- For the first 100 units: $1.5 per unit
- For the next 100 units (101-200): $2.5 per unit
- For units above 200: $3.5 per unit

29. You are given two string arrays `word1` and `word2`. A string array is considered **equivalent** if the strings in the array concatenated form the same string. Return `true` if `word1` and `word2` are equivalent, otherwise return `false`. **Example 1: Input:** word1 = ["ab", "c"], word2 = ["a", "bc"] **Output:** true. **Example 2 :Input:** word1 = ["a", "cb"], **word2 =** ["ab", "c"] **Output:** false.

30. Given an integer array `nums`, find the contiguous subarray (containing at least one number) which has the largest sum and return its sum.**Example1**. **Input**:nums = [-2, 1, -3, 4, -1, 2, 1, -5, 4], **Output:** 6 //The subarray [4, -1, 2, 1] has the largest sum = 6.

31. Given an array of characters, compress it in-place. The length after compression must be the same or smaller than the original array. Compression should be done using the following rules:

- Count consecutive repeating characters and store the character followed by the count.**Example**: **Input: ["a","a","b","b","c","c","c"],Output:["a","2","b","2","c","3"].**

32. Given an array `nums` of n integers where n > 1, return an array output such that output[i] is equal to the product of all the elements of `nums` except

`nums[i]`.**Example:Input:** [1, 2, 3, 4] **Output:** [24, 12, 8, 6].

33.To sort an array containing only `0s`, `1s`, and `2s`, you can use the Dutch National Flag algorithm, which efficiently sorts the array in a single pass. Here's how you can implement it: **Example: Input:** [2, 0, 1, 2, 0, 1, 1] **Output:** [0, 0, 1, 1, 1, 2, 2]

34. Given an array containing `n` distinct numbers taken from `0` to `n`, find the missing number. **Example: Input:** [3, 0, 1], **Output:** 2.

35. Given a string `word` and a character `ch`, reverse the segment of `word` that starts at the beginning and ends at the first occurrence of `ch` (inclusive). If `ch` does not exist in `word`, return `word` unchanged. **Example: Input:** word = "abcdefd", ch = "d", **Output:** "dcbaef"

36. Given a string `s`, reverse the order of characters in each word within a sentence while still preserving whitespace and initial word order. **Example: Input:** `s` = "Let's take LeetCode contest" **Output:** "s'teL ekat edoCteeL tsetnoc".

37. **Check If Array is Monotonic**

Description: An array is considered monotonic if it is either entirely non-increasing or entirely non-decreasing.

Given an array, return `true` if the array is monotonic, otherwise return `false`. **Example1: Input:** nums = [1, 2, 2, 3] **Output:** true, **example2: Input:** nums = [1, 3, 2] **Output:** false

38. **Plus One**

Description: You are given a large integer represented as an array of digits where each digit is in the range [0, 9]. The most significant digit is at the start of the array. Increment the large integer by one and return the resulting array of digits. **Example: Input:** digits = [1, 2, 3] **Output:** [1, 2, 4]

39. Number of Students Doing Homework at a Given Time.

Description: You are given two integer arrays `startTime` and `endTime`, and an integer `queryTime`. The `i-th` student started their homework at the time `startTime[i]` and finished it at the time `endTime[i]`.

Return the number of students doing their homework at `queryTime`. More formally, return the number of students where `queryTime` lies in the interval `[startTime[i], endTime[i]]` inclusive.

Example 1: Input: startTime = [1, 2, 3], endTime = [3, 2, 7], queryTime = 4 **Output:** 1 **Explanation:** Only the third student was doing homework at time 4.

40. You are given a list of strings `sentences`, where each sentence is a string containing words separated by spaces. Your task is to return the maximum number of words found in any single sentence.

Example 1: Input: sentences = ["alice and bob love leetcode", "i think so too", "this is great thanks very much"] **Output:** 6

Explanation: The first sentence has 5 words.

- The second sentence has 4 words.

- The third sentence has 6 words.

Thus, the maximum number of words is 6.

41.You are given an n x n 2D matrix representing an image, rotate the image by 90 degrees (clockwise). You must rotate the matrix **in-place**, which means you cannot use another 2D matrix to accomplish the rotation.

Example:

Input: matrix = [[1,2,3],

[4,5,6],

[7,8,9]]

Output: [[7,4,1],

[8,5,2],

[9,6,3]]

42. Given an integer num, repeatedly add all its digits until the result has only one digit, and return it.**Example**: Input **:num = 38,** Output: **2.**

Explanation:

- **The process is like:**
  - **38 → 3 + 8 = 11**
  - **11 → 1 + 1 = 2**
  - **Since 2 has only one digit, return 2.**