**Basic Questions: -**

**1. Input & Output**

1. Take two numbers as input and print their sum.
2. Take your name as input and print "Hello <name>!".
3. Take a number and print its square.

**2. Conditionals**

1. Check if a number is odd or even.
2. Check if a number is positive, negative, or zero.
3. Given three numbers, print the largest one.
4. Write a program to check if a year is a leap year.

**3. Loops**

1. Print numbers from 1 to 10 using a for loop.
2. Print multiplication table of a number.
3. Print the sum of all numbers from 1 to n.
4. Print Fibonacci sequence up to n terms.

**4. Lists**

1. Take n numbers as input and store in a list.
2. Find the largest and smallest element in a list.
3. Reverse a list without using .reverse() or slicing.
4. Count how many times a number appears in a list.

**5. Strings**

1. Count the number of vowels in a string.
2. Check if a string is a palindrome.
3. Remove all spaces from a string.
4. Replace all vowels in a string with \*.

**6. Functions**

1. Create a function to calculate factorial of a number.
2. Create a function that returns the sum of elements in a list.
3. Create a function to check if a number is prime.

**7. Tuples & Sets**

1. Create a tuple and print its hash.
2. Find the union & intersection of two sets.
3. Remove duplicates from a list using a set.

**8. Dictionaries**

1. Count the frequency of each word in a string.
2. Create a dictionary from two lists: keys and values.
3. Find the key with the maximum value in a dictionary.

**9. List Comprehensions**

1. Create a list of squares from 1 to 10.
2. Create a list of all even numbers between 1 and 20.
3. Create a list of coordinates (x, y) where x is 1–3 and y is 1–2.

**1. Given an integer, , perform the following conditional actions:**

**If is odd, print Weird**

**If is even and in the inclusive range of to , print Not Weird**

**If is even and in the inclusive range of to , print Weird**

**If is even and greater than , print Not Weird**

**Intermediate Questions:-**

**1. Data Structures**

1. Implement a **stack** using a list.
2. Implement a **queue** using collections.deque.
3. Create a program to **merge two sorted lists** into one sorted list.
4. Write a program to find **all unique elements** from a list without using set().

**2. String & Text Processing**

1. Count how many times each **character** appears in a string.
2. Find the **longest word** in a sentence.
3. Check if two strings are **anagrams**.
4. Encode and decode a message using a **Caesar cipher**.

**3. Mathematical Problems**

1. Find all **prime numbers** up to n using the Sieve of Eratosthenes.
2. Find the **GCD** (Greatest Common Divisor) of two numbers without using math.gcd.
3. Create a **decimal to binary converter** without using bin().
4. Generate **Pascal’s Triangle** for n rows.

**4. File Handling**

1. Read a file and count the number of **lines, words, and characters**.
2. Find the **most frequent word** in a text file.
3. Merge the contents of two text files into a new file.

**5. Functions & Modules**

1. Write a **recursive** function to reverse a string.
2. Create a custom Python **module** and import it in another script.
3. Write a decorator to log the **execution time** of a function.

**6. Lists, Tuples, Sets, Dicts**

1. Flatten a nested list without using external libraries.
2. Convert a list of tuples into a dictionary.
3. Sort a dictionary by **values**.
4. Find the **second largest number** in a list.

**7. Problem-Solving**

1. Implement a **basic calculator** supporting +, -, \*, /.
2. Simulate a simple **ATM machine** (deposit, withdraw, check balance).
3. Create a **to-do list manager** with add/remove/display features.

**8. List Comprehensions & Lambdas**

1. Create a list of squares for numbers that are divisible by 3.
2. Use a lambda to sort a list of tuples by the **second element**.
3. Filter out all negative numbers from a list using filter().