Batch - 1

1. Write a method that given an integer array, returns the second largest element.

2. Given a string, return a new string where every delay word “not” followed by “bad” is replaced by “good”. E.g. “This movie is not that bad!” → “This movie is good!”

3. Check if two strings are anagrams of each other (ignoring case and non-letter characters).

4. Given an integer *n*, compute the sum of digits of *n!* (factorial).

5. Given two sorted integer arrays, merge them into one sorted array without using built-in sort functions.

6. Write a program to rotate an array by *k* positions to the right.

Bach - 2

7. Given a string, find the length of the longest substring without repeating characters.

8. Count how many prime numbers are less than *n* and print them.

9. Given an integer, return its reverse. e.g. 12345 → 54321, also handle negative numbers.

10. Given a list of integers, return all unique triplets [i, j, k] such that their sum is zero (no duplicate triplets). (Similar to “3-sum”)

11. Implement a stack class using array and show push, pop and display the content of the stack in a menu driven way.

12. Given a string, check whether it is a palindrome ignoring spaces, punctuation, and case.

Batch - 3

13. Given a positive integer, determine whether it is a power of two without using loops (recursion or bit manipulation).

14. Given two strings, check if one is a rotation of the other. e.g. “waterbottle” is rotation of “erbottlewat”.

15. Given an integer array, find the maximum product of two distinct elements.

16. Given an array and a target sum, find all pairs of numbers that sum to the target (without duplicates).

17. Given a string, return the count of each character in it (case insensitive), ignoring non-letters.

18. Given a number represented by an array of digits, plus one to the number. e.g. [1,2,9] → [1,3,0].

Batch - 4

19. Given two version numbers, compare them. e.g. “1.2.5” vs “1.2.3” → first is bigger.

20. Given a number, print its prime factors.

21. Given a decimal number, convert it to binary string.

22. Given a number *n*, generate all permutations of numbers from 1 to *n*.

23. Given a list of words, find the longest word from the string.

24. Given a string, compress it by replacing consecutive duplicate letters with letter followed by count. e.g. “aaabbc” → “a3b2c1”.