1. **Matrix addition**

   Matrix addition involves adding two matrices element by element. This operation is only valid if both matrices have the same dimensions.

   Input: Gven two matrices A[i][j] and B[i][j] .
   Output: Compute: C[i][j]=A[i][j]+B[i][j]
   Example:
   Let the number of rows and columns of matrix be : 2 2
   The elements of the A[i][j]:
   1 2
   3 4
   The elements of the B[i][j]:
   5 6
   7 8

   Resultant matrix C[i][j] after addition:
    6   8
   10 12

2. **Valid palindrome**

   A phrase is a palindrome if, after converting all uppercase letters into lowercase letters it reads the same forward and backward.

   Given a string s, return true if it is a palindrome, or false otherwise.

   Input: string s= “madam”
   Output : true

   Input: string s= “hello”
   Output : false

3. **Check If Two String Arrays are Equivalent**

   Given two string arrays word1 and word2, return true if the two arrays represent the same string, and false otherwise.

   A string is represented by an array if the array elements concatenated in order forms the string.

   Example 1:
   Input :
    word1 = {"ab", "c"}   word2= {"a", "bc"}

   Output: True The string arrays are equivalent.

   Example 2:

Input: word1 = ["a", "cb"], word2 = ["ab", "c"]
Output: false . The string arrays are not equivalent

4. **String Concatenation and String Length:**

   Input two strings s1 and s2. Concatenate two strings using strcat() and calculate the length of a string without using strlen().

5. **Remove duplicates from sorted array**
   Given an integer array **nums** and an integer **val**, remove all occurrences of **val** in **nums** in-place. The order of the elements may be changed. Then return the number of elements in **nums** which are not equal to **val**.

   Consider the number of elements in nums which are not equal to val be k, to get accepted, you need to do the following things:

   Change the array nums such that the first k elements of nums contain the elements which are not equal to val. The remaining elements of nums are not important as well as the size of nums.

   Return k.

   Example:

   Input: nums = [0,0,1,1,1,2,2,3,3,4]

   Output: 5, nums = [0,1,2,3,4,_,_,_,_,_]

   Explanation: Your function should return k = 5, with the first five elements of nums being 0, 1, 2, 3, and 4 respectively.

   It does not matter what you leave beyond the returned k (hence they are underscores).

6. **Find First and Last Position of an Element in Sorted Array**
   Given an array of integers nums sorted in non-decreasing order, find the starting and ending position of a given target value.

   If target is not found in the array, return [-1, -1].

   Example 1:
   Input: nums = [5,7,7,8,8,10], target = 8
   Output: [3,4]

   Example 2:
   Input: nums = [5,7,7,8,8,10], target = 6
   Output: [-1,-1]

7. **Matrix multiplication**
   Matrix multiplication involves multiplying two matrices. This operation is valid only when the **number of columns in the first matrix** equals the **number of rows in the second matrix**.
   Example:

Input: Matrix A (2×3):
1 2 3
4 5 6
Matrix B (3 × 2):
7   8
9   10
11 12

Output:
Resultant matrix after multiplication:

```
 58    64
139 154
```

8. **Remove element**
   Given an integer array nums and an integer val, remove all occurrences of val in nums in-place. The order of the elements may be changed. Then return the number of elements in nums which are not equal to val.

   Consider the number of elements in nums which are not equal to val be k, to get accepted, you need to do the following things:

   Change the array nums such that the first k elements of nums contain the elements which are not equal to val. The remaining elements of nums are not important as well as the size of nums.
   Return k.

   Example:

   Input: nums = [0,1,2,2,3,0,4,2], val = 2
   Output: 5, nums = [0,1,4,0,3,_,_,_]
   Explanation: Your function should return k = 5, with the first five elements of nums containing 0, 0, 1, 3, and 4.
   Note that the five elements can be returned in any order.

9. **Plus one**
   You are given a large integer represented as an integer array digits, where each digits[i] is the ith digit of the integer. The digits are ordered from most significant to least significant in left-to-right order. The large integer does not contain any leading 0's.
   Increment the large integer by one and return the resulting array of digits.
   Example:

   Input: digits = [1,2,3]
   Output: [1,2,4]
   Explanation: The array represents the integer 123.
   Incrementing by one gives 123 + 1 = 124.
   Thus, the result should be [1,2,4].

10. Find the maximum and minimum element in an array.
    You are given an integer array, where each digit of the array is an integer. The digits are ordered from most significant to least significant in left-to-right order.

    Print the largest and smallest element n the array
    Example:
    Input : The array = [3, 1, 4, 1, 5, 9, 2, 6, 5]
    Output : Maximum = [9] Minimum = [1]

11. Calculate the sum of all elements in an array.

    Initialize an array variable. A variable will hold the sum of the elements of an array.For each element in the array, add it to sum where sum will contain the total sum of the elements.

    Example:
    Input : The array = [4, 5, 9, 1, 6, 5]
    Output : Sum = 30

12. Reverse an array.
    Use two variables. One starts at the beginning (left = 0). The other starts at the end (right = n - 1). Swap the elements at these two pointers using a temporary variable. Move the pointers closer to the center: Increment left. Decrement right. Stop when the pointers meet or cross.

    Example:
    Original array: [1 2 3 4 5]
    Reversed array: [5 4 3 2 1]

13. **Search Insert Position**
    Given a sorted array of distinct integers and a target value, return the index if the target is found. If not, return the index where it would be if it were inserted in order.

    Example 1:

    Input: nums = [1,3,5,6], target = 2
    Output: 1

    Example 2:

    Input: nums = [1,3,5,6], target = 7
    Output: 4

14. **Pascal's Triangle**
    Given an integer numRows, return the first numRows of Pascal's triangle.

    Example:

Input : 5 as the numRows
Output: In Pascal's triangle, each number is the sum of the two numbers directly above it as shown:

```
    1
   1 1
  1 2 1
 1 3 3 1
1 4 6 4 1
```

15. Isomorphic Strings
Given two strings s and t, determine if they are isomorphic.
Two strings s and t are isomorphic if the characters in s can be replaced to get t.
All occurrences of a character must be replaced with another character while preserving the order of characters. No two characters may map to the same character, but a character may map to itself.

Example 1:
Input: s = "egg", t = "add"
Output: true
Explanation:
The strings s and t can be made identical by: Mapping 'e' to 'a'. Mapping 'g' to 'd'.
Example 2:
Input: s = "foo", t = "bar"
Output: false

16. **Valid Anagram**
Given two strings s and t, return true if t is an anagram of s, and false otherwise.
Example 1:
Input: s = "anagram", t = "nagaram"
Output: true

Example 2:
Input: s = "rat", t = "car"
Output: false