1)Given a string s, find the length of the longest

substring

without repeating characters.

Example 1:

Input: s = "abcabcbb"

Output: 3

Explanation: The answer is "abc", with the length of 3.

Example 2:

Input: s = "bbbbb"

Output: 1

Explanation: The answer is "b", with the length of 1.

Example 3:

Input: s = "pwwkew"

Output: 3

Explanation: The answer is "wke", with the length of 3.

Notice that the answer must be a substring, "pwke" is a subsequence and not a substring.

2)Given a string s, return the longest

palindromic

substring

in s.

Example 1:

Input: s = "babad"

Output: "bab"

Explanation: "aba" is also a valid answer.

Example 2:

Input: s = "cbbd"

Output: "bb"

Constraints:

1 <= s.length <= 1000

s consist of only digits and English letters.

3)



Given a string containing digits from 2-9 inclusive, return all possible letter combinations that the number could represent. Return the answer in any order.

A mapping of digits to letters (just like on the telephone buttons) is given below. Note that 1 does not map to any letters.

Example 1:

Input: digits = "23"

Output: ["ad","ae","af","bd","be","bf","cd","ce","cf"]

Example 2:

Input: digits = ""

Output: []

Example 3:

Input: digits = "2"

Output: ["a","b","c"]

4)Given two strings needle and haystack, return the index of the first occurrence of needle in haystack, or -1 if needle is not part of haystack.

Example 1:

Input: haystack = "sadbutsad", needle = "sad"

Output: 0

Explanation: "sad" occurs at index 0 and 6.

The first occurrence is at index 0, so we return 0.

Example 2:

Input: haystack = "leetcode", needle = "leeto"

Output: -1

Explanation: "leeto" did not occur in "leetcode", so we return -1.

5)You are given a string s. The score of a string is defined as the sum of the absolute difference between the ASCII values of adjacent characters.

Return the score of s.

Example 1:

Input: s = "hello"

Output: 13

Explanation:

The ASCII values of the characters in s are: 'h' = 104, 'e' = 101, 'l' = 108, 'o' = 111. So, the score of s would be |104 - 101| + |101 - 108| + |108 - 108| + |108 - 111| = 3 + 7 + 0 + 3 = 13.

Example 2:

Input: s = "zaz"

Output: 50

Explanation:

The ASCII values of the characters in s are: 'z' = 122, 'a' = 97. So, the score of s would be |122 - 97| + |97 - 122| = 25 + 25 = 50.

6)You are given two strings s and t such that every character occurs at most once in s and t is a permutation of s.

The permutation difference between s and t is defined as the sum of the absolute difference between the index of the occurrence of each character in s and the index of the occurrence of the same character in t.

Return the permutation difference between s and t.

Example 1:

Input: s = "abc", t = "bac"

Output: 2

Explanation:

For s = "abc" and t = "bac", the permutation difference of s and t is equal to the sum of:

The absolute difference between the index of the occurrence of "a" in s and the index of the occurrence of "a" in t.

The absolute difference between the index of the occurrence of "b" in s and the index of the occurrence of "b" in t.

The absolute difference between the index of the occurrence of "c" in s and the index of the occurrence of "c" in t.

That is, the permutation difference between s and t is equal to |0 - 1| + |2 - 2| + |1 - 0| = 2.

Example 2:

Input: s = "abcde", t = "edbac"

Output: 12

Explanation: The permutation difference between s and t is equal to |0 - 3| + |1 - 2| + |2 - 4| + |3 - 1| + |4 - 0| = 12.

7)You are given a 0-indexed array of strings words and a character x.

Return an array of indices representing the words that contain the character x.

Note that the returned array may be in any order.

Example 1:

Input: words = ["leet","code"], x = "e"

Output: [0,1]

Explanation: "e" occurs in both words: "leet", and "code". Hence, we return indices 0 and 1.

Example 2:

Input: words = ["abc","bcd","aaaa","cbc"], x = "a"

Output: [0,2]

Explanation: "a" occurs in "abc", and "aaaa". Hence, we return indices 0 and 2.

Example 3:

Input: words = ["abc","bcd","aaaa","cbc"], x = "z"

Output: []

Explanation: "z" does not occur in any of the words. Hence, we return an empty array.

8)You own a Goal Parser that can interpret a string command. The command consists of an alphabet of "G", "()" and/or "(al)" in some order. The Goal Parser will interpret "G" as the string "G", "()" as the string "o", and "(al)" as the string "al". The interpreted strings are then concatenated in the original order.

Given the string command, return the Goal Parser's interpretation of command.

Example 1:

Input: command = "G()(al)"

Output: "Goal"

Explanation: The Goal Parser interprets the command as follows:

G -> G

() -> o

(al) -> al

The final concatenated result is "Goal".

Example 2:

Input: command = "G()()()()(al)"

Output: "Gooooal"

Example 3:

Input: command = "(al)G(al)()()G"

Output: "alGalooG"

9)Example 1:

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

Output: 4

Explanation: There are 4 good pairs (0,3), (0,4), (3,4), (2,5) 0-indexed.

Example 2:

Input: nums = [1,1,1,1]

Output: 6

Explanation: Each pair in the array are good.

Example 3:

Input: nums = [1,2,3]

Output: 0

10)There are n employees in a company, numbered from 0 to n - 1. Each employee i has worked for hours[i] hours in the company.

The company requires each employee to work for at least target hours.

You are given a 0-indexed array of non-negative integers hours of length n and a non-negative integer target.

Return the integer denoting the number of employees who worked at least target hours.

Example 1:

Input: hours = [0,1,2,3,4], target = 2

Output: 3

Explanation: The company wants each employee to work for at least 2 hours.

- Employee 0 worked for 0 hours and didn't meet the target.

- Employee 1 worked for 1 hours and didn't meet the target.

- Employee 2 worked for 2 hours and met the target.

- Employee 3 worked for 3 hours and met the target.

- Employee 4 worked for 4 hours and met the target.

There are 3 employees who met the target.

Example 2:

Input: hours = [5,1,4,2,2], target = 6

Output: 0

Explanation: The company wants each employee to work for at least 6 hours.

There are 0 employees who met the target.