

1. Given an input string (s) and a pattern (p), implement wildcard pattern matching with support for '?' and '*' where:

'?' Matches any single character.

'*' Matches any sequence of characters (including the empty sequence).

The matching should cover the entire input string (not partial).

Example 1:

Input: s = "aa", p = "a"

Output: false

Explanation: "a" does not match the entire string "aa".

Example 2:

Input: s = "aa", p = ""

Output: true

Explanation: '*' matches any sequence.

Example 3:

Input: s = "cb", p = "?a"

Output: false

Explanation: '?' matches 'c', but the second letter is 'a', which does not match 'b'.

2. Given an array arr of size n which contains elements in range from 0 to n-1, you need to find all the elements occurring more than once in the given array. Return the answer in ascending order. If no such element is found, return list containing [-1].

Examples:

Input: n = 4, arr[] = [0,3,1,2]

Output: -1

Explanation: There is no repeating element in the array. Therefore output is -1.

Input: n = 5, arr[] = [2,3,1,2,3]

Output: 2 3

Explanation: 2 and 3 occur more than once in the given array.

3. Given a string str and an integer k, return true if the string can be changed into a pangram after at most k operations, else return false.

A single operation consists of swapping an existing alphabetic character with any other lowercase alphabetic character.

Note - A pangram is a sentence containing every letter in the English alphabet.

Examples :

Input: str = "the quick brown fox jumps over the lazy dog", k = 0

Output: true

Explanation: the sentence contains all 26 characters and is already a pangram.

Input: str = "aaaaaaaaaaaaaaaaaaaaaaaaaaaa", k = 25

Output: true

Explanation: The word contains 26 instances of 'a'. Since only 25 operations are allowed. We can keep 1 instance and change all others to make str a pangram.

Input: str = "a b c d e f g h i j k l m", k = 20

Output: false

Explanation: Since there are only 13 alphabetic characters in this case, no amount of swapping can produce a pangram here.

4. Given two strings A and B, find if A is a subsequence of B.

Example 1:

Input:

A = AXY

B = YADXCP

Output: 0

Explanation: A is not a subsequence of B as 'Y' appears before 'A'.

Input:

A = gksrek

B = geeksforgeeks

Output: 1

Explanation: A is a subsequence of B.