

## Problem Statement

### JUSPAY OA Question: You can Refer [LeetCode 712](#)

A cloud storage service provider is developing a system that helps users synchronize files across multiple devices. One major challenge in file synchronization is resolving differences between two versions of a file that may have been edited independently on different devices. These differences often appear in the form of mismatched strings in text-based data — such as configuration files, code, or plain text documents.

To intelligently merge these file versions, the system must determine the minimal changes needed to make the two strings identical while incurring the least "cost." In this context, each character has an associated cost of deletion equal to its ASCII value. The goal is to remove characters from each string such that the resulting strings become equal and the total ASCII sum of all deleted characters is minimized.

For example:

If  $s1 = \text{"sea"}$  and  $s2 = \text{"eat"}$ , you could delete 's' from  $s1$  (ASCII 115) and 't' from  $s2$  (ASCII 116) to make both equal to "ea", with a total cost of 231.

You are tasked with implementing the core logic of this reconciliation process. Given two strings  $s1$  and  $s2$ , return the lowest possible ASCII sum of characters that must be deleted from the strings to make them equal.

This program will be critical in the file comparison engine, where minimizing data loss and sync time is essential. The implementation must be efficient, as it may be called thousands of times across different parts of a distributed system.

This assessment challenges the candidate's understanding of dynamic programming, optimization under constraints, and practical applications in file and data synchronization.

Constraints:

$1 \leq s1.length, s2.length \leq 1000$

$s1$  and  $s2$  consist of lowercase English letters.

Example 1:

Input:  $s1 = \text{"sea"}, s2 = \text{"eat"}$

Output: 231

Explanation: Deleting "s" from "sea" adds the ASCII value of "s" (115) to the sum.

Deleting "t" from "eat" adds 116 to the sum.

At the end, both strings are equal, and  $115 + 116 = 231$  is the minimum sum possible to achieve this.

Example 2:

Input: s1 = "delete", s2 = "leet"

Output: 403

Explanation: Deleting "dee" from "delete" to turn the string into "let", adds  $100[d] + 101[e] + 101[e]$  to the sum.

Deleting "e" from "leet" adds  $101[e]$  to the sum.

At the end, both strings are equal to "let", and the answer is  $100+101+101+101 = 403$ .

If instead we turned both strings into "lee" or "eet", we would get answers of 433 or 417, which are higher.

Input0:

prepinsta

prep

Output0:

543

Input3:

asdddff

asdd

Output3:

304

Input1:

decode

code

Output1:

201

Input4:

qweer

werr

Output4:

328

Input2:

ghjk

thj

Output2:

326

Input5:

qweweeww

weee

Output5:

470