**Print SCS** ([Shortest Common Supersequence](https://leetcode.com/problems/shortest-common-supersequence/)) - LCS + print LCS

**Supersequence:** Supersequence of two string is a subsequence is which char of both strings can be find.

Example: str1 = “brute”, str2= “groot”

o/p: bgrooute(8)

**Length of supersequence:** To get the length of SCS it sould have common letters once and include all different letters.

=> LCS = 2 (“rt”), n = 5, m = 5

=> length of SCS= n + m - LCS => 5 +5 - 2 = 8

**How to print SCS string?**

**step-1 : make the dp LCS table**

**=>** The task is to print the SCS string, but observe first the table for a case-

text1 = brute, text2 = groot

LCS = “rt”

g  **r** o o  **t**

| - | 0 | 0 | 0 | 0 | 0 | 0 |
| --- | --- | --- | --- | --- | --- | --- |
| b | **0** | 0 | 0 | 0 | 0 | 0 |
| **r** | 0 | 0 | **1** | 1 | 1 | 1 |
| u | 0 | 0 | 1 | 1 | 1 | 1 |
| **t** | 0 | 0 | 1 | 1 | 1 | **2** |
| e | 0 | 0 | 1 | 1 | 1 | 2 (starting point) |

**SCS = “bgrooute”**

Matching: 1 + diagonal value

Not-matching: max(upper row, left column)

**Step-2: Extract the SCS string**

Now our task is to collect the matching characters once and rest all characters such that a supersequence can be made from the table. we can start from the (n, m) and carry a empty string ans = “”; with us

* Take two pointers i = n, j = m
* Matching- take the char and add it to the string, move diagonally up.

i–; j–; take i-1 char

* Not-matching- compare btw dp[i-1] & dp[i][j-1] whichever has the max value move to that place and take its char.

if(dp[i-1][j] > dp[i][j-1]) i–; take i-1 char

else j–; take j-1 char