DP26: Print Longest Common Subsequence - **E:\Interview prep\DEV C++**

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

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

text1 = abcde, text2 = bdgek

b d g e k

| - | 0 | 0 | 0 | 0 | 0 | 0 |
| --- | --- | --- | --- | --- | --- | --- |
| a | **0** | 0 | 0 | 0 | 0 | 0 |
| b | 0 | **1** | 1 | 1 | 1 | 1 |
| c | 0 | 1 | 1 | 1 | 1 | 1 |
| d | 0 | 1 | 2 | 2 | 2 | 2 |
| e | 0 | 1 | 2 | 2 | 3 | 3 (starting point) |

**LCS = “bde”**

Matching: 1 + diagonal value

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

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

Now our task is to collect the matching characters from the table as they are the part of LCS (yellow highlighted) 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–;

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

if(dp[i-1][j] > dp[i][j-1]) i–;

else j–;