

\*solution---

```
-----  
def lcs(X, Y, m, n):  
    L = [[0 for i in range(n+1)] for j in range(m+1)]  
  
    # Following steps build L[m+1][n+1] in bottom up fashion. Note  
    # that L[i][j] contains length of LCS of X[0..i-1] and Y[0..j-1]  
    for i in range(m+1):  
        for j in range(n+1):  
            if i == 0 or j == 0:  
                L[i][j] = 0  
            elif X[i-1] == Y[j-1]:  
                L[i][j] = L[i-1][j-1] + 1  
            else:  
                L[i][j] = max(L[i-1][j], L[i][j-1])  
  
        # Create a string variable to store the lcs string  
        lcs = ""  
  
        # Start from the right-most-bottom-most corner and  
        # one by one store characters in lcs[]  
        i = m  
        j = n  
        while i > 0 and j > 0:  
  
            # If current character in X[] and Y are same, then  
            # current character is part of LCS  
            if X[i-1] == Y[j-1]:  
                lcs += X[i-1]  
                i -= 1  
                j -= 1  
  
            # If not same, then find the larger of two and  
            # go in the direction of larger value  
            elif L[i-1][j] > L[i][j-1]:  
                i -= 1  
  
        else:  
            j -= 1  
  
        # We traversed the table in reverse order  
        # LCS is the reverse of what we got  
        lcs = lcs[::-1]  
        print("LCS of " + X + " and " + Y + " is " + lcs)
```

# Driver program

X = "AGGTAB"

Y = "GXTXAYB"

```
m = len(X)
n = len(Y)
lcs(X, Y, m, n)
```

-----

\*Output----

LCS of AGGTAB and GXTXAYB is GTAB