 <b>Marwadi University</b> Marwadi Chandarana Group	<b>Marwadi University</b> <b>Faculty of Engineering and Technology</b> <b>Department of Information and Communication Technology</b>	
<b>Subject:</b> Design and Analysis of Algorithm	<b>Aim: : Implementing Longest Common Sub-sequence using Dynamic Programming Approach</b>	
<b>Experiment No: 08</b>	<b>Date: 13/09/2025</b>	<b>Enrollment No: 92301733049</b>

## AIM: Implementing Longest Common Sub-sequence using Dynamic Programming Approach

```

#include <iostream>
#include <algorithm>
using namespace std;


// Function to find length of LCS
int LCS(string X, string Y, int m, int n) {
    int dp[m + 1][n + 1];

    // Build the dp table in bottom-up manner
    for (int i = 0; i <= m; i++) {
        for (int j = 0; j <= n; j++) {
            if (i == 0 || j == 0)
                dp[i][j] = 0;
            else if (X[i - 1] == Y[j - 1])
                dp[i][j] = 1 + dp[i - 1][j - 1];
            else
                dp[i][j] = max(dp[i - 1][j], dp[i][j - 1]);
        }
    }

    // dp[m][n] contains length of LCS
    return dp[m][n];
}

// Function to print the actual LCS string
string printLCS(string X, string Y, int m, int n) {
    int dp[m + 1][n + 1];

```

 <b>Marwadi University</b> Marwadi Chandarana Group	<b>Marwadi University</b> <b>Faculty of Engineering and Technology</b> <b>Department of Information and Communication Technology</b>	
<b>Subject:</b> Design and Analysis of Algorithm	<b>Aim: : Implementing Longest Common Sub-sequence using Dynamic Programming Approach</b>	
<b>Experiment No: 08</b>	<b>Date: 13/09/2025</b>	<b>Enrollment No: 92301733049</b>

```

// Build dp table
for (int i = 0; i <= m; i++) {
    for (int j = 0; j <= n; j++) {
        if (i == 0 || j == 0)
            dp[i][j] = 0;
        else if (X[i - 1] == Y[j - 1])
            dp[i][j] = 1 + dp[i - 1][j - 1];
        else
            dp[i][j] = max(dp[i - 1][j], dp[i][j - 1]);
    }
}

// Backtrack to find LCS string
int i = m, j = n;
string lcs = "";


while (i > 0 && j > 0) {
    if (X[i - 1] == Y[j - 1]) {
        lcs = X[i - 1] + lcs;
        i--;
        j--;
    } else if (dp[i - 1][j] > dp[i][j - 1])
        i--;
    else
        j--;
}
return lcs;
}

int main() {
    string X, Y;
    cout << "Enter first string: ";
    cin >> X;
    cout << "Enter second string: ";
    cin >> Y;

    int m = X.length();
    int n = Y.length();

    int length = LCS(X, Y, m, n);

```

 <b>Marwadi University</b> Marwadi Chandarana Group	<b>Marwadi University</b> <b>Faculty of Engineering and Technology</b> <b>Department of Information and Communication Technology</b>	
<b>Subject:</b> Design and Analysis of Algorithm	<b>Aim: : Implementing Longest Common Sub-sequence using Dynamic Programming Approach</b>	
<b>Experiment No: 08</b>	<b>Date: 13/09/2025</b>	<b>Enrollment No: 92301733049</b>

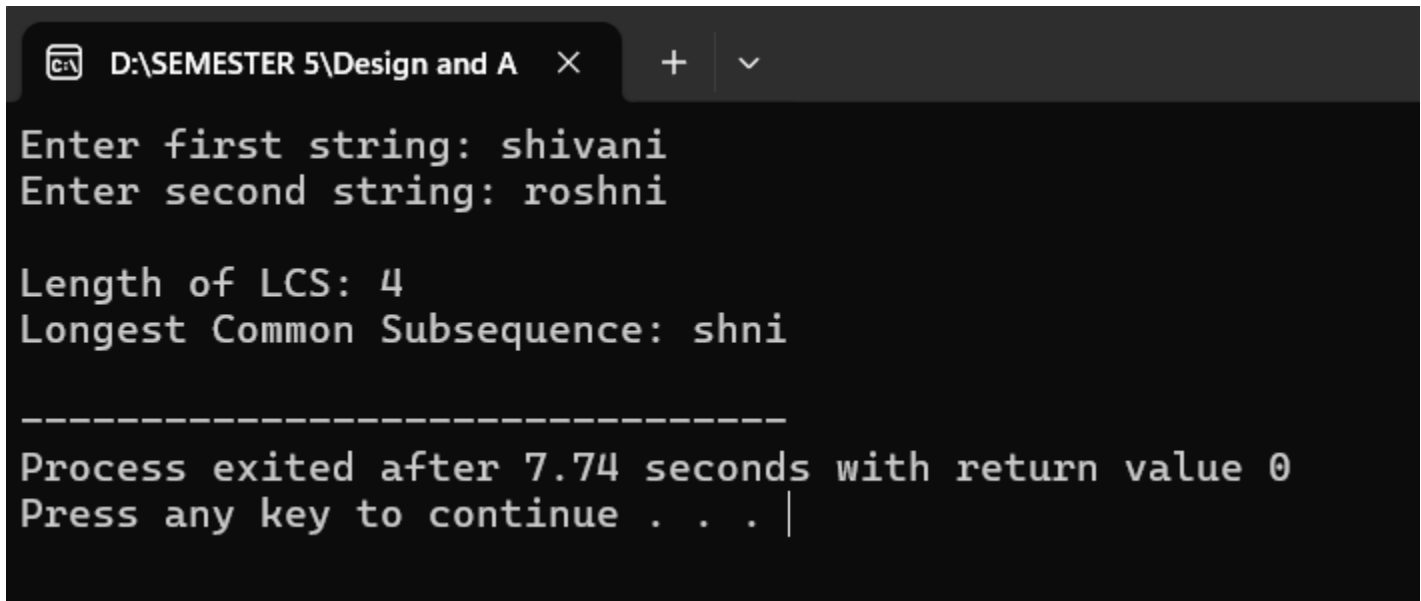
```

string lcsString = printLCS(X, Y, m, n);

cout << "\nLength of LCS: " << length << endl;
cout << "Longest Common Subsequence: " << lcsString << endl;

return 0;
}

```



```

D:\SEMESTER 5\Design and A
Enter first string: shivani
Enter second string: roshni

Length of LCS: 4
Longest Common Subsequence: shni

-----
Process exited after 7.74 seconds with return value 0
Press any key to continue . . .

```