Marwadi University	**Marwari University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Design and Analysis of Algorithms (01CT0512)	**Aim:** Implementing Longest Common Sub-sequence using Dynamic Programming Approach	
Experiment No: 08	**Date:**	**Enrollment No: 92200133030**

Aim: Implementing the Longest Common Sub-sequence using Dynamic Programming Approach

IDE: Visual Studio Code

Longest Common Sub-sequence

Theory: -

- The Longest Common Subsequence (LCS) problem is a classic problem in computer science and bioinformatics that involves finding the longest sequence that can be derived from two given sequences without altering the order of their elements. It is an important problem in string processing and is widely used in applications such as text comparison, DNA sequence analysis, and data diffing tools.

1. Problem Definition

- Given two sequences, the task is to find the longest subsequence that is present in both sequences. A subsequence is a sequence derived by deleting some or no elements from the original sequence without rearranging the order of the remaining elements.
  - For example:
    Sequence 1: "abcde"
    Sequence 2: "ace"
    The LCS is "ace" because it is the longest sequence that appears in the same order in both.

2. Applications of LCS

- Text comparison: Helps detect similarities between text files or documents.
- DNA sequence alignment: Identifies common patterns in genetic sequences.
- Version control systems: Finds differences between versions of files.
- Spell checkers: Matches words with similar spellings.

3. Approach to find LCS :-

- **The most common approach to solving the LCS problem is using Dynamic Programming.**
  1. Define a DP table: Create a table where each cell represents the length of the LCS for two substrings of the input sequences.
  2. Initialize the table: Set the first row and column to 0 since the LCS of any sequence with an empty sequence is 0.
  3. Use a recurrence relation:
     - If the current characters of both sequences match:
       $dp[i][j]=dp[i-1][j-1]+1$
     - If they don't match:

Marwadi University	**Marwari University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Design and Analysis of Algorithms (01CT0512)	**Aim:** Implementing Longest Common Sub-sequence using Dynamic Programming Approach	
Experiment No: 08	**Date:**	**Enrollment No: 92200133030**

dp[i][j]=max(dp[i−1][j],dp[i][j−1])

4. Retrieve the result: The value in the bottom-right cell of the table gives the length of the LCS.

Algorithm: -

Programming Language: - C++

Code :-

```
#include<bits/stdc++.h>
using namespace std;

string Longest_Common_Subsequence(string X, string Y) {
    int m = X.length();
    int n = Y.length();
    int dp[m + 1][n + 1];

    for (int i = 0; i <= m; i++) {
        for (int j = 0; j <= n; j++) {
```

Marwadi University	**Marwari University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Design and Analysis of Algorithms (01CT0512)	**Aim:** Implementing Longest Common Sub-sequence using Dynamic Programming Approach	
Experiment No: 08	**Date:**	**Enrollment No: 92200133030**

```
            if (i == 0 || j == 0) {
                dp[i][j] = 0;
            }
            else if (X[i - 1] == Y[j - 1]) {
                dp[i][j] = dp[i - 1][j - 1] + 1;
            }
            else {
                dp[i][j] = max(dp[i - 1][j], dp[i][j - 1]);
            }
        }
    }

    string lcsString;
    int i = m, j = n;
    while (i > 0 && j > 0) {
        if (X[i - 1] == Y[j - 1]) {
            lcsString = X[i - 1] + lcsString;
            i--;
            j--;
        }
        else if (dp[i - 1][j] > dp[i][j - 1]) {
            i--;
        }
        else {
            j--;
        }
    }

    return lcsString;
}

int main() {

    string X, Y;

    cout << "Enter The First String :- ";
    cin >> X;

    cout << "Enter The Second String :- ";
    cin >> Y;

    string result = Longest_Common_Subsequence(X, Y);
    cout << "Length of LCS is " << result.length() << endl;
    cout << "LCS is: " << result << endl;
```

Marwadi University	**Marwari University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Design and Analysis of Algorithms (01CT0512)	**Aim:** Implementing Longest Common Sub-sequence using Dynamic Programming Approach	
Experiment No: 08	**Date:**	**Enrollment No: 92200133030**

```
    return 0;
}
```

Output :-

```
PS D:\Aryan Data\Usefull Data\Semester - 5\Design-and-Analysis-of-Algorithms\Lab - Manual\Experiment - 8> cd "d:\Aryan Data\Usefull Data\Semester - 5\Design-and-An
alysis-of-Algorithms\Lab - Manual\Experiment - 8\" ; if ($?) { g++ Longest_Common_Subsequence.cpp -o Longest_Common_Subsequence } ; if ($?) { .\Longest_Common_Subs
equence }
Enter The First String :- AGGTAB
Enter The Second String :- GXTXAYB
Length of LCS is 4
LCS is: GTAB
PS D:\Aryan Data\Usefull Data\Semester - 5\Design-and-Analysis-of-Algorithms\Lab - Manual\Experiment - 8>
```

Space Complexity:- _____
Justification: -

Time Complexity:

Best Case Time Complexity: _____
Justification: -

Worst Case Time Complexity:- _____
Justification: -

Conclusion:-