Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Department of Computer Science and Engineering (Data Science)

Experiment – 8 (Longest Common Subsequence)

NAME: DIVYESH KHUNT **SAPID:60009210116**

Aim: Implementation of Longest Common Subsequence using Dynamic approach.

Theory: Given two strings, **S1** and **S2**, the task is to find the length of the longest subsequence present in both of the strings.

We can use the following steps to implement the dynamic programming approach for LCS.

- Create a 2D array **dp[][]** with rows and columns equal to the length of each input string plus 1 [the number of rows indicates the indices of **S1** and the columns indicate the indices of **S2**].
- Initialize the first row and column of the dp array to 0.
- Iterate through the rows of the dp array, starting from 1 (say using iterator **i**).
  - For each **i**, iterate all the columns from **j = 1 to n**:
    - If **S1[i-1]** is equal to **S2[j-1]**, set the current element of the dp array to the value of the element to (**dp[i-1][j-1] + 1**).
    - Else, set the current element of the dp array to the maximum value of **dp[i-1][j]** and **dp[i][j-1]**.

After the nested loops, the last element of the dp array will contain the length of the LCS.

Time Complexity: O(m * n) where m and n are the string lengths.

Example -

X \ Y		a	s	w	v
	0	1	2	3	4
0	0	0	0	0	0
a 1	0	↖ **1**	← 1	← 1	← 1
r 2	0	↑ 1	↑ 1	↑ 1	↑ 1
s 3	0	↑ 1	↖ **2**	← 2	← 2
w 4	0	↑ 1	↑ 2	↖ **3**	← 3
q 5	0	↑ 1	↑ 2	↑ 3	← 3
v 6	0	↑ 1	↑ 2	↑ 3	↖ **4**

Code -

```c
#include <stdio.h>
#include <string.h>
int lcs(char X[], char Y[], int m, int n) {
    int i, j;
    int L[m][n];
    for(i=0; i<m; i++)
    {
        for(j=0; j<n; j++)
        {
            L[i][j]=0;
        }
    }
    for(i=0; i<m; i++)
    {
        for(j=0; j<n; j++)
        {
            printf("%d\t", L[i][j]);
        }
        printf("\n");
    }
    for (int i = 0; i <m; i++) {
        for (int j = 0; j < n; j++) {
            if (i == 0 || j == 0)
                L[i][j] = 0;
            else if (X[i - 1] == Y[j - 1])
                L[i][j] = L[i - 1][j - 1] + 1;
            else
                L[i][j] = (L[i - 1][j] > L[i][j - 1]) ? L[i - 1][j] : L[i][j - 1];
        }
    }
    printf("S1 : %s \nS2 : %s \n", X, Y);
    printf("LCS: %d", L[m-1][n-1]+1);
    printf("\nLCS TABLE - \n");
    for(i=0; i<m; i++)
    {
        for(j=0; j<n; j++)
        {
            printf("%d\t", L[i][j]);
        }
        printf("\n");
    }
}
int main() {
    char S1[100], S2[100];
    printf("Enter the firt string: ");
    scanf("%s", &S1);
    printf("Enter the second string: ");
    scanf("%s", &S2);
    int m = strlen(S1);
    int n = strlen(S2);

    lcs(S1, S2, m, n);

    return 0;
}
```

OUTPUT-

```
Enter the firt string: AMITESH
Enter the second string: DATAMSIENCSH
0       0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0       0
S1 : AMITESH
S2 : DATAMSIENCSH
LCS: 6
LCS TABLE -
0       0       0       0       0       0       0       0       0       0       0
0       0       1       1       1       1       1       1       1       1       1
0       0       1       1       1       2       2       2       2       2       2
0       0       1       1       1       2       2       3       3       3       3
0       0       1       2       2       2       2       3       3       3       3
0       0       1       2       2       2       2       3       4       4       4
0       0       1       2       2       2       3       3       4       4       4       5
```