

<b>Name</b>	<b>Harsh Chandra</b>
<b>UID no.</b>	<b>2021700013</b>
<b>Experiment No.</b>	<b>4</b>

<b>AIM:</b>	<b>Dynamic Programming - Longest Common Subsequence</b>
<b>Program 1</b>	
<b>ALGORITHM/ THEORY:</b>	<p>A subsequence of a string is a sequence that is generated by deleting some characters (possibly 0) from the string without altering the order of the remaining characters. For example, “abc”, “abg”, “bdf”, “aeg”, “acefg”, etc are subsequences of the string “abcdefg”.</p> <pre> m := length(X) n := length(Y) for i = 1 to m do   C[i, 0] := 0 for j = 1 to n do   C[0, j] := 0 for i = 1 to m do   for j = 1 to n do     if xi = yj       C[i, j] := C[i - 1, j - 1] + 1       B[i, j] := 'D'     else       if C[i - 1, j] ≥ C[i, j - 1]         C[i, j] := C[i - 1, j] + 1         B[i, j] := 'U'       else         C[i, j] := C[i, j - 1]         B[i, j] := 'L' return C and B </pre>

**PROGRAM:**

```
#include <stdio.h>
#include <string.h>

int i, j, m, n, LCS_table[20][20];
char b[20][20];
char S1[20], S2[20];
void lcsAlgo() {
    m = strlen(S1);
    n = strlen(S2);

    for (i = 0; i <= m; i++)
        LCS_table[i][0] = 0;
    for (i = 0; i <= n; i++)
        LCS_table[0][i] = 0;

    for (i = 1; i <= m; i++)
        for (j = 1; j <= n; j++) {
            if (S1[i - 1] == S2[j - 1]) {
                LCS_table[i][j] = LCS_table[i - 1][j - 1] + 1;
            } else if (LCS_table[i - 1][j] >= LCS_table[i][j - 1]) {
                LCS_table[i][j] = LCS_table[i - 1][j];
            } else {
                LCS_table[i][j] = LCS_table[i][j - 1];
            }
        }

    int index = LCS_table[m][n]; char lcsAlgo[index + 1];
    lcsAlgo[index] = '\0';
    int i = m, j = n;
    while (i > 0 && j > 0) {
        if (S1[i - 1] == S2[j - 1]) {
            lcsAlgo[index - 1] = S1[i - 1];
            i--;
            j--;
            index--;
        }
        else if (LCS_table[i - 1][j] > LCS_table[i][j - 1])
            i--;
        else
            j--;
    }
}
```

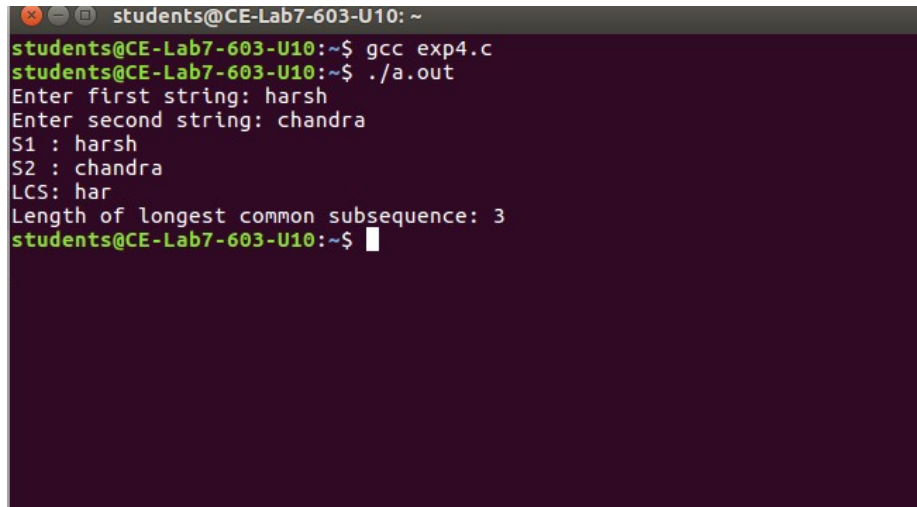
```

}

printf("S1 : %s \nS2 : %s \n", S1, S2);
printf("LCS: %s\n", lcsAlgo);
printf("Length of longest common subsequence: %zu", strlen(lcsAlgo));
}
int main() {
    printf("Enter first string: ");
    scanf("%s",S1);
    printf("Enter second string: ");
    scanf("%s",S2);
    lcsAlgo();
    printf("\n");
}

```

## RESULT:



```

students@CE-Lab7-603-U10: ~
students@CE-Lab7-603-U10:~$ gcc exp4.c
students@CE-Lab7-603-U10:~$ ./a.out
Enter first string: harsh
Enter second string: chandra
S1 : harsh
S2 : chandra
LCS: har
Length of longest common subsequence: 3
students@CE-Lab7-603-U10:~$

```

## CONCLUSION:

Longest common subsequence problem has optimal substructure property as well as overlapping subproblems property. Also I learnt to memoize a solution efficiently.