Name	Siddiqui Hawaiza Sabeel
UID no.	2022701010
Experiment No.	4
SUBJECT	DAA

AIM:	**Implement Dynamic Programming - Longest Common Subsequence**
Program 1	
ALGORITHM/ THEORY:	The longest common subsequence (LCS) is defined as the longest subsequence that is common to all the given sequences, provided that the elements of the subsequence are not required to occupy consecutive positions within the original sequences. **Time Complexity:** $O(n * 2^n)$ • Let us count the total subsequences with lengths 1, 2, . . . , n-1, n. • From theory of permutation and combination we know number of combinations with 1 element is ${}^nC_1$. Number of combinations with 2 elements are ${}^nC_2$ and so on. • So a string of length n has ${}^nC_1 + {}^nC_2 + \ldots {}^nC_n = 2^n-1$ different possible subsequences. • Each subsequence takes O(n) time to compare. **Auxiliary Space:** O(n) As we can reuse the same string.

PROGRAM:

```
#include <stdio.h>
#include <string.h>
#include <time.h>
int lcs[100][100];
int max(int x, int y)
{
   if (x > y)
      return x;
   else
      return y;
}
int LCS(char a[], char b[], int la, int lb)
{
   if (la == 0 || lb == 0)
   {
      lcs[la][lb] = 0;
      return 0;
   }
   if (a[la - 1] == b[lb - 1])
   {
      lcs[la][lb] = 1 + LCS(a, b, la - 1, lb - 1);
      return 1 + LCS(a, b, la - 1, lb - 1);
   }
   else
   {
      lcs[la][lb] = max(LCS(a, b, la - 1, lb), LCS(a, b, la, lb - 1));
      return lcs[la][lb];
   }
}
int main()
{
   char a[100], b[100];
   double t1 = clock();
   printf("\nEnter first string : ");
   scanf("%s", a);
   printf("\nEnter second string : ");
   scanf("%s", b);
   int t = LCS(a, b, strlen(a), strlen(b));
   t1 = clock() - t1;
   t1 = t1 / CLOCKS_PER_SEC;
   int i = strlen(a), j = strlen(b);
   char c[t + 1];
   c[t] = '\0';
   while (i > 0 && j > 0)
```

```
  {
    if (a[i - 1] == b[j - 1])
    {
      c[t - 1] = a[i - 1];
      i--;
      j--;
      t--;
    }
    else if (lcs[i - 1][j] > lcs[i][j - 1])
      i--;
    else
      j--;
  }
  printf("\nThe Longest Common Subsequence is %s", c);
}
```

RESULT:

```
PS C:\Users\91913\Desktop\SPIT\SEM 4\DAA> gcc exp_4.c
PS C:\Users\91913\Desktop\SPIT\SEM 4\DAA> ./a.exe

 Enter first string : ABCBDAB

 Enter second string : BDCABA

 The Longest Common Subsequence is BDAB
PS C:\Users\91913\Desktop\SPIT\SEM 4\DAA>
```

CONCLUSION:	I understood the dynamic programming approach to find the Longest Common Subsequence between 2 strings. I learnt that The dynamic programming method to solve LCS is better than the Recursive method to solve LCS