Name	**Rahul Chalwadi**
UID no.	**2021700012**
Experiment No.	**4**

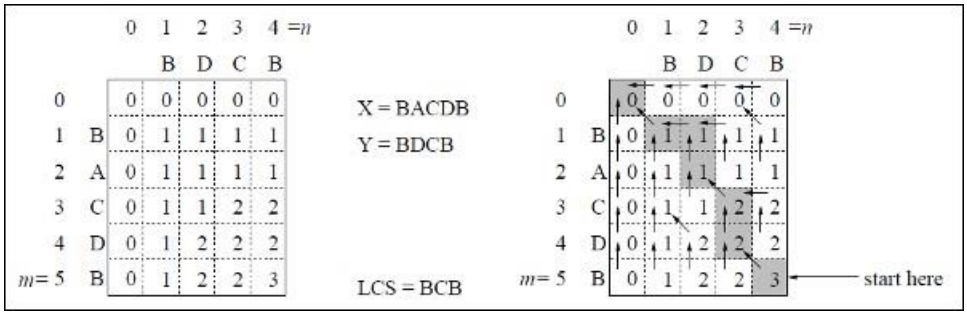
AIM:	**Dynamic Programming – Longest Common Subsequence**
Program 1	
PROBLEM STATEMENT :	**Implement dynamic programming approach to find out the longest common subsequence in two given strings**

ALGORITHM/ THEORY:

function lcs(X, Y, m, n):

```
 L[m+1][n+1] = {0}


 for i = 0 to m do
 for j = 0 to n do
 if i == 0 or j == 0 then
 L[i][j] <- 0
 else if X[i-1] == Y[j-1] then
 L[i][j] <- L[i-1][j-1] + 1
 else
 L[i][j] <- max(L[i-1][j], L[i][j-1])  lcs = ""

 i <- m

 j <- n
 while i > 0 and j > 0 do
 if X[i-1] == Y[j-1] then

 lcs <- X[i-1] + lcs
 i <- i - 1
 j <- j - 1
 else if L[i-1][j] > L[i][j-1] then
 i <- i - 1
 else
 j <- j - 1

 return struct {L[m][n], lcs}
```

Example:



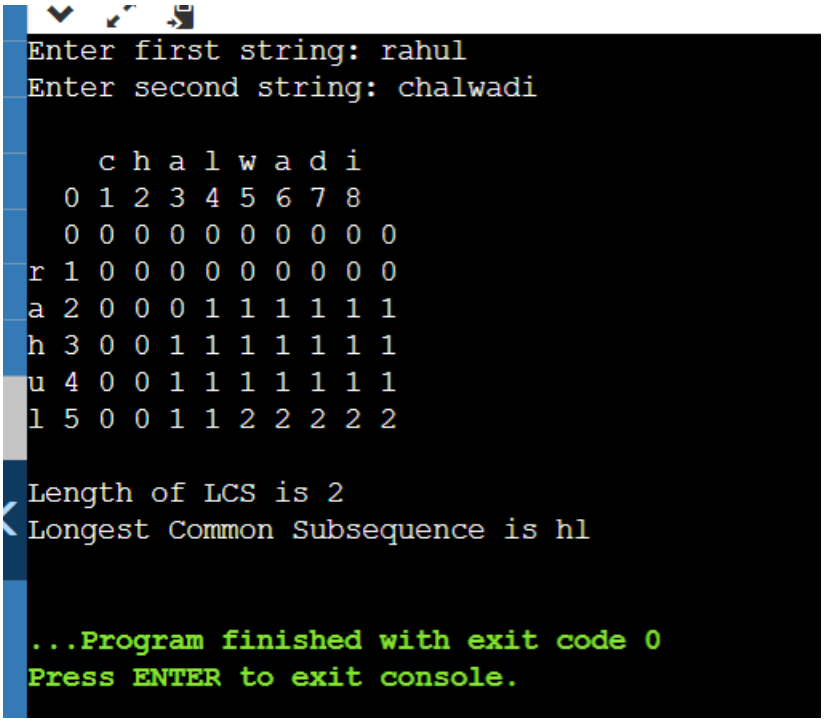
Program:

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
 typedef struct
lcs_output
{ int
length;
char *subsequence;
} lcs_output;
 lcs_output lcs(char X[], char Y[], int m, int
n)
{
 int L[m + 1][n +
1];
 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];
}
}

int length = L[m][n]; char
lcs[length + 1];
lcs[length] = '\0';

int i = m, j = n; while
(i > 0 && j > 0)
```

```
{
 if (X[i - 1] == Y[j - 1])
{
lcs[length - 1] = X[i - 1]; i--; j--; length--;
}

else if (L[i - 1][j] > L[i][j - 1]) i--; else
j--;
}

printf("\n "); for (int j = 0; j < n; j++)
{ printf("%c ", Y[j]);
}
printf("\n "); for (int j = 0; j <= n; j++)
{ printf("%d ", j);
}
printf("\n"); for (int i = 0; i <= m; i++)
{
printf("%c %d ", i > 0 ? X[i - 1] : ' ', i); for (int j = 0; j <= n; j++)
{ printf("%d ", L[i][j]);
} printf("\n");
} printf("\n");

lcs_output l1; l1.length = L[m][n]; l1.subsequence = lcs; return l1;
}

int main()
{ char *s1 = malloc(100), *s2 = malloc(100);
```

```
printf("Enter first string: ");
scanf("%s", s1); printf("Enter
second string: "); scanf("%s",
s2);

int m = strlen(s1); int
n = strlen(s2);

lcs_output l = lcs(s1, s2, m, n);
printf("Length of LCS is %d\n", l.length); printf("Longest
Common Subsequence is %s\n", l.subsequence);

return 0;
}
```

RESULT:



Conclusion: Thus, in this experiment, I implemented the dynamic programming approach to find out the longest common subsequence in two given strings successfully.