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

int i, j, m, n, LCS_table[20][20];
char S1[20] = "ACADB", S2[20] = "CBDA", b[20][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);
  int len = strlen(lcsAlgo);
  printf("length is : %d",len);
}

int main() {
  lcsAlgo();
  printf("\n");
}
```

Time Complexity: O(nm)

Output: