#include <iostream>

#include <string>

using namespace std;

// Defining Directions

#define None 0

#define Left 1

#define Top 2

#define Diagonal 3

// Function to return the maximum value between two integers

int max(int a, int b) { return (a > b) ? a : b; }

// Function to return which of the two integers is larger

int maxNo(int a, int b) { return (a >= b) ? 1 : 2; }

void longest\_common\_subsequence(string str1, string str2) {

// Initializing variables to store the length of the strings

int len1 = str1.length();

int len2 = str2.length();

// Initialize a 2D array to store the length of longest common subsequence and the directions

int C[len2 + 1][len1 + 1];

int D[len2 + 1][len1 + 1];

// Initialize an empty string to store the longest common subsequence

string subsequence = "";

// Fill the 2D array using dynamic programming to find the length of longest common subsequence

for(int i = 0; i < len2 + 1; i++) {

for(int j = 0; j < len1 + 1; j++) {

if(i == 0 || j == 0) {

C[i][j] = 0;

D[i][j] = None;

}

else if(str2[i-1] == str1[j-1]) {

C[i][j] = 1 + C[i-1][j-1];

D[i][j] = Diagonal;

}

else if(str2[i-1] != str1[j-1]) {

C[i][j] = max(C[i][j-1], C[i-1][j]);

if(maxNo(C[i][j-1], C[i-1][j]) == 1){

D[i][j] = Left;

}

else{

D[i][j] = Top;

}

}

}

}

// print the two-dimensional array of the lengths of the longest common subsequences

for(int i = 0; i < len2 + 1; i++) {

for(int j = 0; j < len1 + 1; j++) {

cout << C[i][j] << " ";

}

cout << endl;

}

int i=len2;

int j=len1;

int direction;

// get the longest common subsequence by tracing back through the direction of each element

do{

direction = D[i][j];

switch(direction){

case None:

break;

case Left:

j--;

break;

case Top:

i--;

break;

case Diagonal:

subsequence = str1[j-1] + subsequence;

i--;

j--;

}

}while(direction != None);

cout << "Length of the longest common subsequence is : " << C[len2][len1] << endl;

cout << "The longest common subsequence is : " << subsequence << endl;

}

int main() {

string str1, str2;

cout << "Enter the first string : ";

getline(cin, str1);

cout << "Enter the second string : ";

getline(cin, str2);

longest\_common\_subsequence(str1, str2);

return 0;

}

OUTPUT:

