#include <iostream>

#include <vector>

using namespace std;

string longestCommonSubstring(string s1, string s2) {

int m = s1.length();

int n = s2.length();

// Create a 2D table to store the lengths of common substrings

vector<vector<int>> dp(m + 1, vector<int>(n + 1, 0));

int maxLength = 0; // Length of the longest common substring

int endIndexS1 = 0; // Ending index of the longest common substring in s1

// Fill the dp table

for (int i = 1; i <= m; i++) {

for (int j = 1; j <= n; j++) {

if (s1[i - 1] == s2[j - 1]) {

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

if (dp[i][j] > maxLength) {

maxLength = dp[i][j];

endIndexS1 = i - 1;

}

}

}

}

// Extract the longest common substring

string result = s1.substr(endIndexS1 - maxLength + 1, maxLength);

return result;

}

int main() {

string s1, s2;

cout << "Enter the first string: ";

cin >> s1;

cout << "Enter the second string: ";

cin >> s2;

string commonSubstring = longestCommonSubstring(s1, s2);

cout << "Longest common substring: " << commonSubstring << endl;