#include <iostream>

#include <vector>

#include <string>

using namespace std;

string longestCommonSubstring(string s1, string s2) {

int m = s1.length();

int n = s2.length();

// Initialize a 2D vector to store the lengths of common substrings

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

// Variables to store the length of the longest common substring and its ending position

int maxLength = 0;

int endIndex = 0;

// 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];

endIndex = i - 1; // Update the ending position of the common substring

}

} else {

dp[i][j] = 0; // Reset the length of common substring

}

}

}

// Extract the longest common substring

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

return result;

}

int main() {

string s1 = "ABAB";

string s2 = "BABA";

// Find the longest common substring

string longestSubstring = longestCommonSubstring(s1, s2);

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

return 0;

}