```cpp
#include <bits/stdc++.h>
const int INF = 1e9;
const int MOD = 1e9+7;
const long long LINF = 1e18;
#define dump(x)  cout << 'x' << ' = ' << (x) << ` `;
#define FOR(i,a,b) for(int i=(a);i<(b);++i)
#define REP(i,n) for(int i=0;i<(n);++i)
#define REPR(i,n) for(int i=n;i>=0;i--)
#define FOREACH(x,a) for(auto& (x) : (a) )
typedef long long ll;
using namespace std;
constexpr int MAX_NUM = 5001;

int dp[MAX_NUM][MAX_NUM];

string LCS(string &s, string &t) {
    string ret;

    // dpテーブルを初期化
    REP(i,MAX_NUM) REP(j,MAX_NUM) dp[i][j] = 0;

    // LCSを求める
    REP(i,s.length()) {
        REP(j,t.length()) {
            if (s[i] == t[j]) dp[i+1][j+1] = max(dp[i+1][j+1], dp[i][j]+1);
            else dp[i+1][j+1] = max(dp[i+1][j], dp[i][j+1]);
        }
    }

    // 解の復元をする
    int ci = s.length(); int cj = t.length();

    while (true) {
        if (ci == 0 || cj == 0) break;

        if(s[ci-1] == t[cj-1] and dp[ci-1][cj-1]+1 == dp[ci][cj]) {
      ret.push_back(s[ci-1]);
      ci--,cj--;
    } else if(dp[ci-1][cj] == dp[ci][cj]){
      ci--;
    } else if(dp[ci][cj - 1] == dp[ci][cj]){
      cj--;
    }
    }

    reverse(ret.begin(), ret.end());
    return ret;
}

int main(int argc, char const *argv[]) {
    string s,t;
    cin >> s >> t;
    string ans = LCS(s,t);
    cout << ans << endl;
    return 0;
}

```