#include<bits/stdc++.h>

using namespace std;

//-------------sir method-----------

string lcs(string &s1, string &s2, int n, int m){

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

for(int i=0;i<=n;i++){

dp[i][0] = 0;

}

for(int j=0;j<=m;j++){

dp[0][m] = 0;

}

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

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

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

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

}

else{

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

}

}

}

int i=n,j=m;

string res = "";

while(i>0 && j>0){

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

res = s1[i-1] + res;

i--;

j--;

}

else{

if(dp[i-1][j]>dp[i][j-1]){

i--;

}

else{

j--;

}

}

}

return res;

}

//---------------------------my method----------

string lcs(string &s1, string &s2, int n, int m){

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

for(int i=0;i<=n;i++){

dp[i][0] = "";

}

for(int j=0;j<=m;j++){

dp[0][m] = "";

}

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

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

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

dp[i][j] = dp[i-1][j-1] + s1[i-1];

}

else{

string l = dp[i-1][j];

string r = dp[i][j-1];

if(l.size()>=r.size()){

dp[i][j] = l;

}

else{

dp[i][j] = r;

}

}

}

}

return dp[n][m];

}

string lcs(string &s1, string &s2, int n, int m, vector<vector<string> > &dp){

if(n==0 || m==0){

return dp[n][m] = "";

}

if(dp[n][m]=="$"){

if(s1[n-1]==s2[m-1]){

dp[n][m] = lcs(s1,s2,n-1,m-1,dp) + s1[n-1];

}

else{

string l = lcs(s1,s2,n-1,m,dp);

string r = lcs(s1,s2,n,m-1,dp);

if(l.size()>=r.size()){

dp[n][m] = l;

}

else{

dp[n][m] = r;

}

}

}

return dp[n][m];

}

int main(){

string s1,s2;

cin>>s1>>s2;

int n = s1.size();

int m = s2.size();

vector<vector<string> > dp(n+1,vector<string> (m+1,"$"));

cout<<lcs(s1,s2,n,m);

}