#include<bits/stdc++.h>

using namespace std;

// easy way

bool isSubsequence(string s, string t) {

int i=0,j=0;

while(j<t.size()){

if(s[i]==t[j]){

i++;

}

j++;

}

if(i==s.size()){

return true;

}

return false;

}

// my method

int subseq(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 i=0;i<=m;i++){

dp[0][i] = 1;

}

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

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

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

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

dp[i][j] = 1;

}

else{

dp[i][j] = 0;

}

}

else{

if(dp[i-1][j]==1 && dp[i][j-1]==1){

dp[i][j] = 1;

}

else{

dp[i][j] = 0;

}

}

}

}

return dp[n][m];

}

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

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

return dp[n][m] = 1;

}

if(n==0){

return dp[n][m] = 1;

}

if(m==0){

return dp[n][m] = 0;

}

if(dp[n][m]==-1){

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

if(subseq(s1,s2,n-1,m-1,dp)){

return dp[n][m] = 1;

}

}

else{

if(subseq(s1,s2,n-1,m,dp)==1 && subseq(s1,s2,n,m-1,dp)==1){

return dp[n][m] = 1;

}

}

}

return dp[n][m];

}

int main(){

string s1,s2;

cin>>s1>>s2;

int n = s1.size();

int m = s2.size();

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

if(subseq(s1,s2,n,m)==1){

cout<<"true";

}

else{

cout<<"false";

}

}