**Common Child**

A string is said to be a child of a another string if it can be formed by deleting 0 or more characters from the other string. Given two strings of equal length, what's the longest string that can be constructed such that it is a child of both?

For example, ABCD and ABDC have two children with maximum length 3, ABC and ABD. They can be formed by eliminating either the D or C from both strings. Note that we will not consider ABCD as a common child because we can't rearrange characters and ABCD  ABDC.

**Function Description**

Complete the *commonChild* function in the editor below. It should return the longest string which is a common child of the input strings.

commonChild has the following parameter(s):

* *s1, s2*: two equal length strings

**Input Format**

There is one line with two space-separated strings,  and .

**Constraints**

* All characters are upper case in the range ascii[A-Z].

**Output Format**

Print the length of the longest string , such that  is a child of both  and .

**Sample Input**

HARRY

SALLY

**Sample Output**

2

**Explanation**

The longest string that can be formed by deleting zero or more characters from  and  is , whose length is 2.

**Sample Input 1**

AA

BB

**Sample Output 1**

0

**Explanation 1**

 and  have no characters in common and hence the output is 0.

**Sample Input 2**

SHINCHAN

NOHARAAA

**Sample Output 2**

3

**Explanation 2**

The longest string that can be formed between  and  while maintaining the order is .

**Sample Input 3**

ABCDEF

FBDAMN

**Sample Output 3**

2

**Explanation 3**  
 is the longest child of the given strings.

#include<bits/stdc++.h>

#define pb push\_back

#define ll long long

#define INF 1e18

#define vec vector<int>

#define REP(i,a,b) for(i=a;i<b;i++)

using namespace std;

int main()

{

  ios\_base::sync\_with\_stdio(false);

  cin.tie(NULL);

  cout.tie(NULL);

  ll int t=1;

  //cin>>t;

  while(t--)

  {

    string s1,s2;

    cin>>s1>>s2;

    int n=s1.length(),m=s2.length();

    vector<vector<int>> dp;

    int i,j;

    vector<int> x;

    for(i=0;i<n+1;i++)

        x.pb(0);

    dp.pb(x);

    dp.pb(x);

    for(i=1;i<=n;i++)

    {

      for(j=1;j<=m;j++)

      {

       if(s1[i-1]==s2[j-1])

            dp[1][j]=1+dp[0][j-1];

       else

            dp[1][j]=max(dp[0][j],dp[1][j-1]);

      }

      dp[0]=dp[1];

      dp[1]=x;

    }

    cout<<dp[0][m];

  }

}