## 1159 Common Subsequence

### 一、题目

#### 问题描述

A subsequence of a given sequence is the given sequence with some elements (possible none) left out. Given a sequence X = <x1, x2, …, xm> another sequence Z = <z1, z2, …, zk> is a subsequence of X if there exists a strictly increasing sequence <i1, i2, …, ik> of indices of X such that for all j = 1,2,…,k, xij = zj. For example, Z = <a, b, f, c> is a subsequence of X = <a, b, c, f, b, c> with index sequence <1, 2, 4, 6>. Given two sequences X and Y the problem is to find the length of the maximum-length common subsequence of X and Y. The program input is from a text file. Each data set in the file contains two strings representing the given sequences. The sequences are separated by any number of white spaces. The input data are correct. For each set of data the program prints on the standard output the length of the maximum-length common subsequence from the beginning of a separate line.

#### 输入数据

#### 输出数据

#### 输入样例

abcfbc abfcab programming contest  
abcd mnp

#### 输出样例

4 2 0

#### 题目来源

HDU 1159 http://acm.hdu.edu.cn/showproblem.php?pid=1159

### 二、题解

#### 解题思路

动态规划经典例题把一个大的问题分解为多个子问题。

（1）如果ai-1 == bj-1，则当前最长公共子序列为“a0, a1, …, ai-2”与“b0, b1, …, bj-2”的最长公共子序列与am-1的和。长度为“a0, a1, …, ai-2”与“b0, b1, …, bj-2”的最长公共子序列的长度+1。

（2）如果ai-1 != bj-1，则最长公共子序列为max（“a0, a1, …, ai-2”与“b0, b1, …, bj-1”的公共子序列，“a0, a1, …, ai-1”与“b0, b1, …, bj-2”的公共子序列）

当a[i]==b[j]时 dp[i] [j]=dp[i-1] [j-1]+1 当a[i]!=b[j]时 dp[i] [j]=max(dp[i] [j],dp[i-1] [j])

#### 参考程序

#include<stdio.h>  
#include<string.h>  
int max(int a,int b)  
{  
 return a>b?a:b;  
}  
char a[1002],b[1002];  
int dp[1002][1002];  
int main() {  
  
 while(scanf("%s %s",a+1,b+1)!=EOF) {  
 memset(dp,0,sizeof(dp));  
 int maxx=-1;  
 int alen=strlen(a+1);  
 int blen=strlen(b+1);  
 for(int i=1; i<=alen; i++) {  
 for(int j=1; j<=blen; j++) {  
 if(a[i]==b[j])dp[i][j]=dp[i-1][j-1]+1;  
 else dp[i][j]=max(dp[i-1][j],dp[i][j-1]);  
 //maxx=max(maxx,dp[i][j]);  
 }  
 }  
 printf("%d\n",dp[alen][blen]);  
 memset(a,'\0',sizeof(a));  
 memset(b,'\0',sizeof(b));  
 }  
 return 0;  
}

#### 复杂度分析

无

#### 编程技巧

无