**实验一**

**1.解决问题：计算两个字符串或数值组成的序列的最长公共子序列**

**2.运用算法：动态规划算法**

动态规划应用于找出最优子结构的问题，最优子结构既局部最优解能决定全局最优解。

**3.算法实现：**

（一） 首先，创建两个序列：

Xi=<x1,...,xi>,表明X序列的前i个字符，(1<=i<=m)

Yj=<y1,...,yj>,既为Y序列的前j个字符，(1<=j<=n)

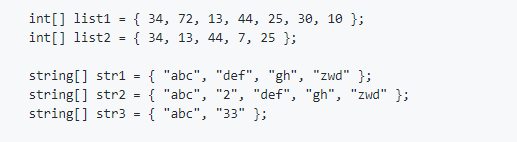
（二）在X、Y序列中找出最长公共子序列，假设最长公共子序列是为Zk=<zi,...,zk>，Zk的长度为k：

1）假设X序列的最后一个字符xm和Y序列的最后一个字符yn相等，既xm=yn:用反证法可以证明，Zk(最长公共子序列)的最后一个字符zk=xm=yn，因为Zk∈LCS(X,Y) , 所以Zk-1∈LCS（Xm-1,Yn-1),Z的前缀Zk-1是Xi-1序列和Yj-1序列的最长公共子序列。所以Zk的长度等于LCS（Xi-1,Yj-1）的长度加1

2）假设X序列的最后一个字符xm和Y序列的最后一个字符yn不相等，既xm≠yn:用反证法可以证明，Zk要么Z∈LCS(Xm-1, Y)，要么Zk∈LCS(X , Yn-1)。由于zk≠xm与zk≠yn其中至少有一个必成立，若zk≠xm则有Z∈LCS(Xm-1 , Y)，以此类推如果zk≠yn 则有Z∈LCS(X , Yn-1)。所以问题变成了求LCS(Xm-1 , Y)和 LCS(X , Yn-1)的长度。在这两种情况下，需要选择长度最长的序列，既max{LCS(Xm-1 , Y)的长度, LCS(X , Yn-1)的长度}。

（三）输出最长公共子序列

**4、输入：**



**5、输出：**

