

实验五

✎ 编程题 1 (总分: 100.00)

#	题目	分值
1	实验5-1: 最长公共子序列 Description 一个给定序列的子序列是在该序列中删去若干元素后得到的序列。给定两个序列X和Y, 当另一序列Z既是X的子序列又是Y的子序列时, 称Z是序列X和Y的公共子序列。例如, 若X={A, B, C, B, D, B, A}, Y={B, D, C, A, B, A}, 则序列{B, C, A}是X和Y的一个公共子序列, 但它不是X和Y的一个最长公共子序列。序列{B, C, B, A}也是X和Y的一个公共子序列, 它的长度为4, 而且它是X和Y的一个最长公共子序列, 因为X和Y没有长度大于4的公共子序列。最长公共子序列问题就是给定两个序列 $X=\{x_1, x_2, \dots, x_m\}$ 和 $Y=\{y_1, y_2, \dots, y_n\}$, 找出X和Y的一个最长公共子序列。 Input 输入包含多组测试数据。第一行为一个整数C, 表示有C组测试数据, 接下来有C行数据, 每组测试数据占1行, 它由2个给定序列的字符串组成, 两个字符串之间用空格隔开。 Output 你的输出应该有C行, 即每组测试数据的输出占一行, 它是计算出的最长公共子序列长度。 Sample Input 1 ABCBDDBA BDCABA Sample Output 4 参考答案:	100.00

```
#include <iostream>
#include <cstring>
using namespace std;

int max(int a, int b) {
    return (a > b) ? a : b;
}

int lcsLength(string X, string Y) {
    int m = X.length();
    int n = Y.length();
    int L[m + 1][n + 1];

    for (int i = 0; i <= m; i++) {
        for (int j = 0; j <= n; j++) {
            if (i == 0 || j == 0)
                L[i][j] = 0;
            else if (X[i - 1] == Y[j - 1])
                L[i][j] = L[i - 1][j - 1] + 1;
            else
                L[i][j] = max(L[i - 1][j], L[i][j - 1]);
        }
    }

    return L[m][n];
}

int main() {
    int C;
    cin >> C;

    while (C--) {
        string X, Y;
        cin >> X >> Y;
        cout << lcsLength(X, Y) << endl;
    }

    return 0;
}
```

