```cpp
/*** 1110 - An Easy LCS:
 Given two non-empty strings. You have to find the lexicographically
smallest Longest Common Subsequence(LCS) between them.
If the LCS length is 0 then just print ':('.
***/
#include<bits/stdc++.h>
using namespace std;
string a,b;
int dp[105][105], nexta[105][27], nextb[105][27], last[105][27];
int fun(int i,int j){
    if(i==a.size() || j==b.size())return 0;
    if(dp[i][j]!=-1) return dp[i][j];

    int ret=0;
    if(a[i]==b[j]) ret = 1+fun(i+1,j+1);
    else ret = max(fun(i+1,j), fun(i,j+1));

    return dp[i][j] = ret;
}
int main(){
    int tt; scanf("%d",&tt);
    for(int ks=1; ks<=tt; ks++){
        cin >> a >> b;
        a = "+" + a; b = "-" + b;

        memset(dp,-1,sizeof(dp));
        int cnt = fun(0,0);

        if(cnt==0){
            printf("Case %d: :(\n",ks); continue;
        }

        for(int i=0; i<26; i++) last[a.size()][i]=a.size();
        for(int i=a.size()-1; i>=0; i--){
            for(int j=0; j<26; j++){
                nexta[i][j]=last[i+1][j];
                last[i][j]=last[i+1][j];
            }
            int id  = a[i]-'a';
            last[i][id]=i;
        }

        for(int i=0; i<26; i++)last[b.size()][i]=b.size();
        for(int i=b.size()-1; i>=0; i--){
            for(int j=0; j<26; j++){
                nextb[i][j]=last[i+1][j];
                last[i][j]=last[i+1][j];
            }
            int id  = b[i]-'a';
            last[i][id]=i;
        }

        string ans;
        int pa=0,pb=0; int i=0, j=0;
        for(int k=cnt; k>=0; k--){
            for(char ch='a'; ch<='z'; ch++){
                int id=ch-'a';
                pa = nexta[i][id]; pb = nextb[j][id];

                if(a[pa]==b[pb] && dp[pa][pb]==k){
                    ans += ch; i=pa, j=pb;
                    break;
                }
            }
        }
        cout << "Case " << ks << ": " << ans << endl;
    }
}
```