Acwing-4382：打字

芭芭拉是一个速度打字员。

为了检查她的打字速度，她进行了一个速度测试。

测试内容是给定她一个字符串 I，她需要将字符串正确打出。

但是，芭芭拉作为一个速度打字员，在追求速度的同时，难免会发生一些错误，按错一些按键。

最终，芭芭拉打出的字符串为 P。

现在，芭芭拉想知道，能否仅通过删除一些额外字母的方式，将字符串
P 变为字符串
I。

如果可以，则输出需要删除的字母数量，如果不行，则输出 `IMPOSSIBLE`。

输入格式

第一行包含整数
T，表示共有
T 组测试数据。

每组数据占两行，第一行包含字符串
I，第二行包含字符串
P。

输出格式

每组数据输出一个结果，每个结果占一行。

结果表示为 `Case #x: y`，其中
x 为组别编号（从
1 开始），
y 为需要删除的字母数量或 `IMPOSSIBLE`。

数据范围

$1 \le T \le 100$,
字符串 I 和 P 均只包含大小写字母。
$1 \le |I|, |P| \le 10^5$。

输入样例1:

```
2
aaaa
aaaaa
bbbbb
bbbbc
```

输出样例1:

```
Case #1: 1
Case #2: IMPOSSIBLE
```

输入样例2:

```
2
Ilovecoding
IIllovecoding
KickstartIsFun
kkickstartiisfun
```

输出样例2:

```
Case #1: 2
Case #2: IMPOSSIBLE
```

基本思路是两个字符串各开一个指针，`l`表示字符串`I`的指针，`r`表示字符串`P`的指针。

- Case1：`I[l]==P[r]`则两个指针都自增，开始比较后一位
- Case2：`I[l]!=P[r]`则`r++`，表示删除该位。

case2的合理性在于，即使前后出现了重复的符合片段，如`I="abcde"`，`P="abcfffabcde"`，不需要考虑abc取前面的还是后面的，因为在这种逻辑下取了前面的，继续匹配必然将`fffabc`删除。

设置判断的边界调节：

- 如果`l`指针已经到达`I`的末尾⬅️END，此时如果`r`还没有到达`P`的末尾，则需要删除`r`后面的所有字符。
- 如果`r`指针已经到达`P`的 末尾，但是`l`还没有到达`I`的末尾，即再怎么删也删不出理想的造型，那么就判定为不可达。

Code：

```cpp
//
//  main.cpp
//  4382-快速打字
//
//  Created by MacBook Pro on 2023/7/16.
//

#include <iostream>
using namespace std;

int T;
string I,P;

int main() {
    cin>>T;
    for(int i=0;i<T;i++){
        cin>>I>>P;
        if(P.size()<I.size()){
```

```
            cout<<"Case #"<<i+1<<": "<<"IMPOSSIBLE"<<endl;
            continue;
        }
        //双指针开始遍历
        int l=0,r=0;
        bool ans=false;
        int res=0;
        while(true){
            if(I[l]==P[r]){
                //匹配，就直接后移一位
                l++; r++;
            }
            else{
                //不匹配，删去一位r
                res++;
                r++;
            }
            //判断是否结束
            if(l>=I.size()){
                ans=true;
                if(r<P.size()){
                    res+=(P.size()-r);
                }
                break;
            }
            else if(l<I.size()&&r>=P.size()){
                //不能满足
                ans=false;
                break;
            }
        }
        if(ans) cout<<"Case #"<<i+1<<": "<<res<<endl;
        else cout<<"Case #"<<i+1<<": "<<"IMPOSSIBLE"<<endl;
    }
}
```

代码提交状态：Accepted