

# AcWing-4122：字符串转换

给定一个由小写字母构成的字符串  $S$ 。

再给定一个由若干个各不相同的小写字母按字典序排序构成的字符串  $F$ 。

现在，你可以对字符串  $S$  进行字符串转换操作。

每次选中其中一个字符（即某个小写字母），将其转换为一个按照字母顺序与其相邻（上一个或下一个）的小写字母。

例如，  
 $c$  可以转换为  
 $b$  或  
 $d$ 。

额外的，我们将按照循环顺序考虑字母，即我们认为  
 $a$  的上一个字母为  
 $z$ ，  
 $z$  的下一个字母为  
 $a$ 。

请问，至少需要进行多少次操作，可以使得字符串  
 $S$  中的每个字母都出现在字符串  
 $F$  中。

### 输入格式

第一行包含整数  
 $T$ ，表示共有  
 $T$  组测试数据。

每组数据第一行包含一个字符串  $S$ 。

第二行包含一个字符串  $F$ 。

### 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），  
 $y$  为最少操作次数。

### 数据范围

$1 \leq T \leq 100$ ，  
 $1 \leq |S| \leq 10^5$ ，  
 $1 \leq |F| \leq 26$ 。

## 输入样例1：

```
2
abcd
a
pppp
p
```

时/空限制：	3s / 64MB
总通过数：	893
总尝试数：	1839
来源：	Google Kickstart2021 Round H Problem A
算法标签	▼

## 输出样例1:

```
Case #1: 6
Case #2: 0
```

## 输入样例2:

```
3
pqrst
ou
abd
abd
aaaaaaaaaaaaaab
aceg
```

## 输出样例2:

```
Case #1: 9
Case #2: 0
Case #3: 1
```

本题的就是简单的枚举问题，遍历每一个 `s` 字符串中的字符，找26个字母中最接近且在 `f` 中的字符即可。

Code:

```
//
//  main.cpp
//  4122-字符串转换
//
//  Created by MacBook Pro on 2023/8/7.
//

#include <iostream>
#include <algorithm>
#include <string>
#include <climits>
using namespace std;

int main()
{
    int T;
    cin >> T;
    for (int cases = 1; cases <= T; cases++)
    {
        string s, f;
        cin >> s >> f;
```

```
int cnt = 0;
for (int i = 0; i < int(s.length()); i++)
{
    int ans = INT_MAX;
    for (int j = 0; j < int(f.length()); j++){
        ans=min(ans,min(abs(s[i]-f[j]),26-abs(s[i]-f[j])));
    }
    cnt += ans;
}
printf("Case #%d: %d\n", cases, cnt);
}
return 0;
}
```

代码提交状态: Accepted