

C 老和尚的真言——Officially

题目描述

空即是色，色即是空，阿弥陀佛……

这一周，老和尚外出了，留下一卷真言给小和尚参悟。小和尚翻开这一卷真言，善哉，善哉……

这一卷真言，居然是用英文写的！而且还加了密！幸好小和尚很快就看出了门道，他需要做的，只是先把其中全部的“**esolc**”删掉后，再将删除后得到的字符串中所有的“**abc**”换成“**dnalsi**”即可。正巧，你在小和尚这里打下手，于是他来找你帮忙。

输入

第一行包含一个整数 $t(1 \leq t \leq 100)$ ，为数据的组数。

接下来 t 行，每行一个字符串（可能含空格），为加密的真言。

输出

对于每个字符串，输出一行，为处理后的结果。

输入样例

```
2
abc close is a good menesolctor!
aesolcabc
```

输出样例

```
dnalsi close is a good mentor!
dnalsi
```

解题分析

原本以为大家对字符串的学习已经足够应付这样的题目，但是没想到课堂上讲的并不是非常深入，所以造成了这种情况。助教们表示十分抱歉。

基本内容在教材的第十八章与第十六章，这里提供本题使用的几个库函数：

`cin.get()`；——读入一个字符，类比 `getchar()`

`getline(cin,a)`——其中 `a` 为 `string` 类型。读入带有空格的字符串，经常需要与 `cin.get()` 结合使用。

`cin.getline(a,num)`——其中 `a` 为 `char*` 类型，`num` 为 `a` 数组的元素个数（`sizeof(a)` 返回的是 `a` 的字节数，不同）同样是读入带有空格的字符串，也经常需要与 `cin.get()` 结合使用

`gets(a)`——其中 `a` 为 `char*` 类型，同上，C 语言中读入带空格字符串的函数。

`cin>>a`；——`a` 为 `string` 类型或 `char*` 类型，该语句读入不包含空格的字符串。

若 `a` 为 `string` 类型：

`a.length()`——返回字符串 `a` 的长度，若 `a` 包含 `n` 个字符，则返回值为 `n`。

`a.size()`——同上

`a.replace(start,length,b)`——其中 `start` 为整形，为替换的起始位置，`length` 为整形，为替换的长度，`b` 为替换的字符串。该函数的作用即为将 `a` 中起始位置为 `start`，长度为 `length` 的一个子串替换为字符串 `b`。注意该函数既有返回值，即处理后的结果，又会对 `a` 进行处理。

`a.substr(start,length)`——`start` 与 `length` 的意义同上，为取 `a` 中起始位置为 `start`，长度为 `length` 的子串，返回值即为子串，用法：`b=a.substr(start,length);`

`a.erase(start,length)`——`start` 与 `length` 的意义同上，为删除 `a` 中起始位置为 `start`，长度为 `length` 的一段子串。注意该函数既有返回值，即处理后的结果，且也会对 `a` 进行处理。

`a.find('c')/a.find("abc")`——重载函数，前者为在 `a` 串中寻找某一字符，返回值为第一次出现该字符的位置；后者为在 `a` 串中寻找某一子串，返回值为第一次出现该子串的位置，如果没有则返回 `a.npos`（一个较大的正整数）

如果 `a` 是 `char*`

`strlen(a)`——C 语言中计算字符串长度的函数，请注意使用该函数时一定要将其赋给一个变量，从而减少不必要的计算（诸如 `for(int i=0;i<strlen(a);i++)` 的循环实际上时间复杂度是 $O(n^2)$ 大家可以考虑一下为什么）

`strcmp(a,b)`——比较两字符串的字典序大小，返回值记不太清了，大家可以编写程序测试一下（计算机导论的索引制作可以写一个程序进行排序，其中由于是对字符串进行字典序排序，因此这个函数将占主导地位）

此外还有 `strcpy`、`strcat` 等经典函数，请大家自行研究，这里不作赘述。

参考代码：

```
#include<iostream>
#include<string>
using namespace std;
int main()
{
    int T;
    cin>>T;cin.get();
    while(T--){
        string a;
        getline(cin,a);
        while(a.find("esolc")!=a.npos)
            a.erase(a.find("esolc"),5);
        while(a.find("abc")!=a.npos)
            a.replace(a.find("abc"),3,"dnalsi");
        cout<<a<<endl;
    }
}
```