

# 3194 字符串消除

## 问题描述:

给定一个由大写字母'A'、'B'、'C'构成的字符串s，按如下进行消除过程：

1、字符串s中连续相同字母组成的子串，如果子串的长度大于1，那么这些子串会被同时消除，余下的字符拼成新的字符串。

例如："ABCCBCCCAA"中"CC","CCC"和"AA"会被同时消除，余下"AB"和"B"拼成新的字符串"ABB"。

2、反复进行上述消除，直到新的字符串中相邻字符都不相同为止。

例如："ABCCBCCCAA"经过一轮消除得到"ABB"，再经过一轮消除得到"A"。

假设在对字符串s消除开始前，允许在s中任意位置(第一个字符之前、最后一个字符之后以及相邻两个字符之间)插入任意一个字符('A','B'或者'C')，得到字符串t，然后对字符串t经过一系列消除。

请问该如何插入字符，使得字符串t中被消除掉的字符总数（包括插入的字符）最多？

# 3194 字符串消除

- 问题分析

- 定义函数`del`计算字符串中被消除掉的字符总数。
- 把字符‘A’,‘B’,‘C’分别插入在字符串`s`的第一个字符之前、相邻两个字符之间以及最后一个字符之后得到新的字符串`s'`，计算字符串`s'`中被消除掉的字符总数，找到最大值。

# 3194 字符串消除

- 1、定义函数`del(s)`计算字符串中被消除掉的字符总数
  - 从左到右扫描字符串`s`，连续相同字母组成的子串删除，没有被消除的字符保存在另一个字符数组`t`中。
  - `count`=字符串`s`的长度 - 字符串`t`的长度，`count`为这一轮被消除掉的字符个数。
  - 如果字符串`s`和`t`相等，那说明相邻字符都不相同，消除完成，返回0。否则，递归调用函数`del(t)`对字符串`t`开启新一轮消除，被消除的字符总数为：  
`count+del(t)`

```

int del(char *ostr) //ostr指向字符串，函数返回值为被消除的字符个数
{
    char str[MAXN],t[MAXN];
    strcpy(str, ostr);
    int slen = strlen(str), lent = 0, i, l = 0;

    //slen为原始字符串的长度，lent为消除后剩余字符串的长度
    if (slen == 0) return 0;
    str[slen] = '$'; //方便处理最后字符，在字符串的最后额外增加一个特殊字符$
    str[slen + 1] = '\0';
    for(i = 1; i < slen + 1; i++)
    {
        if (str[i] != str[i - 1]) //相邻字符不相等,未被消除的字符保存在数组t中
        {
            if (l == i-1) t[lent++] = str[i-1];
            l = i;
        }
    }
    t[lent] = '\0';
    if (slen == lent) return 0; //两个字符串相等，消除完成，返回0。
    return slen - lent + del(t);

    //递归调用del(t)，被消除的字符总数为：slen-lent+字符串t被消除的字符个数
}

```

# 3194 字符串消除

2、把字符‘A’,‘B’,‘C’分别插入在字符串s的第一个字符之前、相邻两个字符之间以及最后一个字符之后得到新的字符串s’，计算字符串s’中被消除掉的字符总数，找到最大值。

```
char s[MAXN], temp[MAXN];
```

```
scanf("%s", s);
```

```
int i, len = strlen(s), ans = 0;
```

```
for(i = 0; i <= len; i++)
```

```
// 插入在字符串s的第一个字符之前、相邻两个字符之间以及最后一个字符之后
```

```
{
```

```
    strncpy(temp, s, i); temp[i] = '\0'; strcat(temp, "A"); strcat(temp, s + i);
```

```
    //插入字符‘A’
```

```
    ans = MAX(ans, del(temp));
```

```
    strncpy(temp, s, i); temp[i] = '\0'; strcat(temp, "B"); strcat(temp, s + i);
```

```
    //插入字符‘B’
```

```
    ans = MAX(ans, del(temp));
```

```
    strncpy(temp, s, i); temp[i] = '\0'; strcat(temp, "C"); strcat(temp, s + i);
```

```
    //插入字符‘C’
```

```
    ans = MAX(ans, del(temp));
```

```
}
```

```
printf("%d\n", ans);
```