

实现字符串的原地逆置

原地逆置题目要求为不使用额外的链表节点

注意，不适用额外的结点意味着不能分配新的结点空间，但可以使用新的指针

这里给出有头节点和没有头节点的实现方式

没有头节点

```
//假设结构体类型为node
node * reverse(node * head){
    if(!head) {
        return NULL;
    }//没有返回意味着至少有一个结点
    node * newhead,*p,*q;
    newhead = head;
    p = head -> next;
    while(p){
        q = p;
        p = p->next;//把原链表剩下的部分的第一个结点取下来
        q -> next = newhead;
        newhead = q;//头插法
    }
    return newhead;
}
```

有头节点

```
node * reverse(node *head){
    node * p,q;
    p = head -> next;
    head -> next = NULL;
    while(p){
        q = p;
        p = p-> next;
        q ->next = head -> next;
        head -> next = q;
    }
    return head;
}
```

replace函数

这里假设存储方式为字符数组

假设我们实现了一个在给定字符串str中找一个子串sub的第一次出现的位置的函数find,函数原型在下面给出。注意，只需要找到sub第一次出现的位置就行。

find函数的实现可以参考习题课讲义（是某次的一道作业题）

```
int find(char *str, char *sub);
//找到返回下标pos，没找到则返回-1

void replace(char *str, char dest[], char* sub_1, char *sub_2 ){
    //用sub_2代替str中的所有的sub_1
    sub_1_len = strlen(sub_1);
    sub_2_len = strlen(sub_2);
    char *tmp;
    while((pos = find(str, sub_1)) != -1){
        tmp = &str[pos + sub_1_len];
        strcpy(&str[pos], sub_2); //由于题目要求sub_2_len <= sub_1_len
        strcat(str, tmp);
    }
    strcpy(dest, str);
}
```