

串和数组 第 20 题

问题描述:

将用单链表存储的串中所有字符 c 替换成字符 s。

1: 函数结构设计

函数名	Replace
函数正常输入	SqString 型串指针
函数正常输出	替换指定字符并返回 TRUE=1

2: 测试样例设计

	输入	预测结果
一般正常情况	串指针，要替换的字符和替换成的字符	替换完成，返回 TRUE=1
异常情况	串指针为空	返回 FALSE=-1

3: 伪代码描述

如果输入的串指针为空，返回 FALSE=-1
循环遍历串中的每个元素，如果为'c'，则替换为's'
遍历完所有元素，返回 TRUE=1

4: 程序描述

```
// 串和数组 第 20 题
```

```
#include "stdafx.h"  
#include <assert.h>
```

```

#include <iostream>
using namespace std;

#define TRUE      1
#define FALSE    -1

typedef char ElemType;

typedef struct LinkList
{
    ElemType dat;
    LinkList *next;
} LinkList;

typedef LinkList *SqString;

// 创建空链表
LinkList *List_InitEmpty(void)
{
    LinkList *head = new LinkList;
    assert(head != NULL);
    head->next = NULL;
    return head;
}

// 尾插法创建链表
LinkList *List_InitRear(ElemType *dat, int n)
{
    int i;
    if(dat == NULL) {
        return List_InitEmpty();
    }
    LinkList *head = new LinkList, *New, *Last;
    assert(head != NULL);
    Last = head;
    for(i = 0; i < n; ++i) {
        Last->next = New = new LinkList;
        assert(New != NULL);
        New->dat = dat[i];
        Last = New;
    }
    Last->next = NULL;
    return head;
}

```

// 串的初始化

```
SqString SqStringInit(ElemType *s)
{
    int n = 0;
    ElemType *p = s;
    while(*p++) {
        ++n;
    }
    return List_InitRear(s, n);
}
```

// 替换指定元素

```
int Replace(SqString lst, ElemType src, ElemType dst)
{
    if(lst == NULL) {
        return FALSE;
    }
    while(lst != NULL) {
        if(lst->dat == src) {
            lst->dat = dst;
        }
        lst = lst->next;
    }
    return TRUE;
}
```

// 输出全部元素

```
void PrintAll(LinkList *lst)
{
    if(lst == NULL) {
        return;
    }
    lst = lst->next;
    while(lst != NULL) {
        cout << lst->dat;
        lst = lst->next;
    }
    cout << endl;
}
```

```

int main(void)
{
    ElemType *str = "TectTectTest";
    SqString s = SqStringInit(str);           // 初始化串
    PrintAll(s);
    cout << Replace(NULL, 'c', 's') << endl; // 异常情况：返回 FALSE=-1
    cout << Replace(s, 'c', 's') << endl;    // 正常情况：替换字符并返回 TRUE=1
    PrintAll(s);

    system("pause");
    return 0;
}

```

5: 结果展示

```

TectTectTest    // 替换前的串
-1              // 异常情况：返回 FALSE=-1
1              // 正常情况：替换字符并返回 TRUE=1
TestTestTest    // 替换后的串

```