



# 南昌大学实验报告

学生姓名： 丁俊      学 号： 8003119100      专业班级： 信安 193 班

实验类型： ☒ 验证 ☐ 综合 ☐ 设计 ☐ 创新      实验日期： 4.8      实验成绩：         

## 一、实验项目名称

栈与字符串

## 二、实验目的

熟练掌握栈与字符串的链式存储及操作

## 三、实验任务

- 1、利用顺序栈结构，编写算法函数 `void Doto16(int m)` 实现十进制无符号整数 `m` 到十六进制数的转换。
- 2、设计算法函数 `void delstring(linkstring s,int i,int len)` 在字符串 `s` 中删除从第 `i` 个位置开始，长度为 `len` 的子串。
- 3、字符串采用带头结点的链表存储，编写函数 `linkstring index(linkstring s,linkstring t)` 查找子串 `s` 在主串 `t` 中第一次出现的位置，若匹配不成功，则返回 `NULL`。

## 四、主要仪器设备及耗材

**Dec++5.15    windows10**

## 五、实验步骤

### 一、十进制到十六进制转换

```
#include "seqstack.h"
void Dto16(int m) {
    sequence_stack s;
    init(&s);
    printf("十进制%d 对应的十六进制是:", m);
    while (m) {
        push(&s, m % 16);
        m = m / 16;
    }
    while (!empty(s)) {
        putchar(read(s) >= 10 ? pop(&s) + 55 : pop(&s) + 48);
    }
    printf("\n");
}
int main() {
    int m;
    printf("请输入带转换的十进制数:\n");
    scanf("%u", &m);
    Dto16(m);
    return 0;
}
```

以下是 pop() 函数，返回栈顶元素，且弹出栈顶元素。

```
void push(sequence_stack *st, datatype x) {
    if (st->top == MAXSIZE) {
        printf("\nThe sequence stack is full");
        exit(1);
    }
    st->a[st->top] = x;
    st->top++;
}

int pop(sequence_stack *st) {
    if (st->top == 0) {
        printf("\nThe sequence stack is empty");
        exit(1);
    }
    return st->a[--st->top];
}
```

2、在字符串 s 中删除从第 i 个位置开始，长度为 len 的子串。

Linkstring.h 头文件：

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
typedef struct node {  
    char data;  
    struct node *next;  
} linkstrnode;  
typedef linkstrnode *linkstring;
```

```
linkstring create() {  
    char ch;  
    linkstring s, head, r;  
    head = r = (linkstring)malloc(sizeof(linkstrnode));  
    head->next = NULL;  
    printf("请输入一个字符串:\n");  
    while ((ch = getchar()) != '\n') {  
        s = (linkstrnode *) malloc(sizeof(linkstrnode));  
        s->data = ch;  
        r->next = s;  
        r = s;  
    }  
    r->next = NULL;  
    return head;  
}
```

```
void print(linkstring s) { // s 是一个指针  
    linkstring p = s->next; // 起始指针  
    while (p) {  
        printf("%c", p->data);  
        p = p->next;  
    }  
    printf("\n");  
}
```

```
void delList(linkstring s) {  
    linkstring p = s;  
    while (s) {  
        free(p);  
        p = s->next;  
    }  
}
```

主代码:

```
#include "linkstring.h"
```

```
void delstring(linkstring s, int i, int len) {
    linkstring p, q;
    int k = 1;
    p = s->next;
    while (p && k < i - 1) { // 待删除的子串的起始结点的前驱结点
        p = p->next;
        k++;
    }
    if (!p || !p->next)
        return;
    else {
        k = 0;
        while (p->next && k < len) { // 循环 len 的长度
            q = p->next;
            p->next = q->next;
            free(q);
            k++;
        }
    }
}

int main() {
    linkstring str;
    str = create();
    print(str);
    delstring(str, 2, 3);
    print(str);
    delList(str);
    return 0;
}
```

### 3、查找子串 s 在主串 t 中第一次出现的位置

```
#include "linkstring.h"

// 带头结点
linkstring index(linkstring s, linkstring t) { // t 是子串
    linkstring p, p1, t1;
    p = s->next;
    while (p) { // 依次循环主串
        p1 = p;
        t1 = t->next; // 每次都要从头循环子串
        while (p1 && t1 && p1->data == t1->data) {
            p1 = p1->next;
            t1 = t1->next;
        }
        if (t1 == NULL)
            return p; // 说明子串在主串中已经遍历到了
        p = p->next;
    }
    return NULL;
}

int main() {
    linkstring s, t, p = NULL;
    s = create();
    t = create();
    print(s);
    print(t);
    p = index(s, t);
    if (p) {
        printf("匹配成功,首次匹配的结点值为:%c\n", p->data);
    } else
        printf("匹配失败\n");
    delList(s);
    delList(t);
    return 0;
}
```

## 六、实验数据以及处理结果

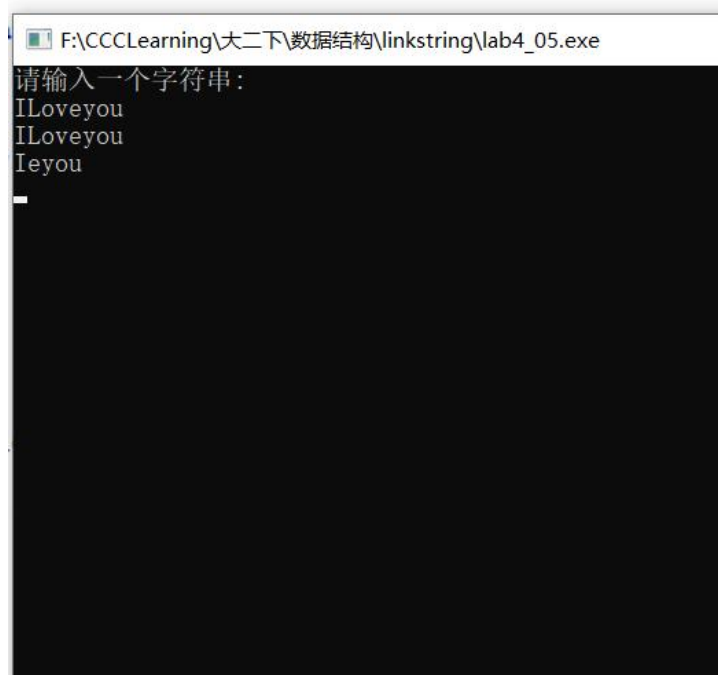
### 1、进制转换

```
F:\CCCLearning\大二下\数据结构\day3\eg1.exe
请输入带转换的十进制数:
15
十进制15对应的十六进制是:F
-----
Process exited after 3.404 seconds with return value 0
请按任意键继续. . .
```

```
F:\CCCLearning\大二下\数据结构\day3\eg1.exe
请输入带转换的十进制数:
31
十进制31对应的十六进制是:1F
-----
Process exited after 0.9485 seconds with return value 0
请按任意键继续. . .
```

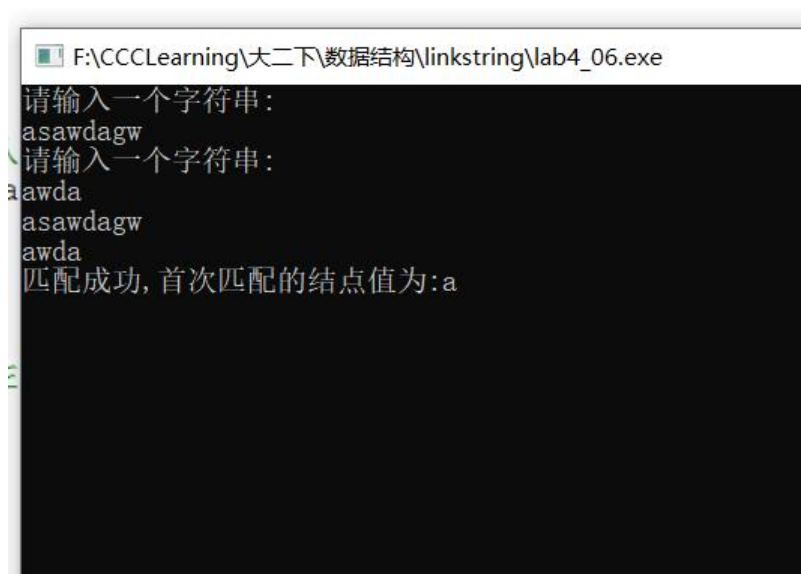
```
F:\CCCLearning\大二下\数据结构\day3\eg1.exe
请输入带转换的十进制数:
125
十进制125对应的十六进制是:7D
-----
Process exited after 3.083 seconds with return value 0
请按任意键继续. . .
```

2、删除指定子串，从第二个字符开始删除长度为 3 的子串




```
F:\CCCLearning\大二下\数据结构\linkstring\lab4_05.exe
请输入一个字符串:
Iloveyou
Iloveyou
Ieyou
```

3、查找子串出现的位置，返回第一个字符，例如在“asawdagw”中查找“awda”子串，返回第一个字母“a”。



```
F:\CCCLearning\大二下\数据结构\linkstring\lab4_06.exe
请输入一个字符串:
asawdagw
请输入一个字符串:
awda
asawdagw
awda
匹配成功, 首次匹配的结点值为:a
```

当不存在该子串时，匹配失败。



```
xt.  
F:\CCCLearning\大二下\数据结构\linkstring\lab4_06.exe  
请输入一个字符串:  
qwertyuio  
请输入一个字符串:  
qwertr  
qwertyuio  
qwertr  
匹配失败
```

## 七、思考讨论题或体会或对改进实验的建议

通过这次实验，我初步掌握了字符串链式的存储和栈先进后出的妙用，解决了一些基础问题和算法。

## 八、参考资料

数据结构 C 语言版