

实验四 堆栈实验

【实验目的】

- (1) 掌握堆栈的两种不同的存储结构。
- (2) 掌握应用堆栈表示数据、并进行有关算法设计的方法。

【实验背景】

堆栈是一种重要且常用的数据结构，其逻辑结构与顺序表、链表相同，但其运算较顺序表、链表有更多的限制，故也称其为运算受限的线性表。

同顺序表的存储结构类似，我们可以用地址连续的存储空间依次存储栈中数据元素，并记录当前栈顶数据元素的位置，这样的栈称为**顺序栈**。

由于入栈和出栈运算都是在栈顶进行，而栈底位置是固定不变的，我们可以将栈底位置设置在向量的起始处；栈顶位置是随入栈和出栈操作而变化的，故需用一个整型变量 **top** 来记录当前栈顶元素在向量中的位置。类似顺序表的类型定义，顺序栈的类型可以定义如下：

```
#define maxlen 100
typedef struct{
    Datatype data[maxlen];
    int top;
}SeqStack;
```

与链表的存储结构类似，我们也用链式存储结构存储堆栈的数据元素 $a_i, a_{i+1} (i=1, 2, \dots, n)$ ，并记录当前栈顶数据元素的位置，这样的栈称为**链栈**。

链栈中的节点同链表结点一样，也包括两个域：**data** 域是数据域，用来存放数据元素 a_i 的值；**next** 域是指针域，用来存放 a_i 的直接后继 a_{i+1} 的存储地址。节点的类型可以描述为：

```
typedef struct node{
    datatype data;
    struct node *next;
}LinkStack;
```

由于栈只能在栈顶进行插入和删除(入栈和出栈)操作，而链表只有在表头进行这样的操作时才最为简单，故在链栈中头节点为栈顶，尾节点为栈底。如图 6.1 所示，LS 是 LinkStack 类型的指针变量，可以有如下定义：

```
LinkStack *LS;
```

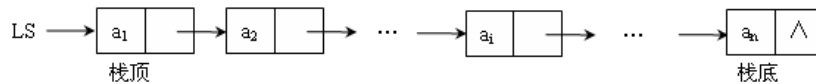


图 6.1 链栈示意图

顺序栈和链栈的基本运算同顺序表和链表，只是栈的插入与删除操作都只在栈顶进行。

【实验任务】

1、程序验证 分别建立含有若干个元素的顺序栈和链栈，并分别实现顺序栈和链栈的入栈和出栈操作。

2、算法填空 将十进制正整数转换成十六进制数的算法如下。完善该算法，并通过运行来验证。

```
void dectohex(long num) {
```

```

SeqStack s;
InitStack(&s);
while(num) {
    int k=num%16;
    Push(&s,k);
    _____;
}
while(!StackEmpty(&s)) {
    int x=GetTop(&s);
    if(x<10)
        printf("%d ",x);
    else {
        switch( _____ ) {
            case 10: printf("A ");break;
            case 11: printf("B ");break;
            case 12: printf("C ");break;
            case 13: printf("D ");break;
            case 14: printf("E ");break;
            case 15: printf("F ");break;
        }
    }
    Pop(&s);
}
printf("\n");
}

```

3、算法设计

(1) 编写算法，利用栈判断所给字符串是否具有中心对称关系。要求用尽可能少的时间完成判断。

提示：①字符串的中心对称是如 *xyzyx* 和 *abcedcba* 的形式；

②字符串可以考虑顺序存储或链接存储；

③判断该字符串是否中心对称，可以将一半字符先入栈。

(2) 设计算法判断一个算术表达式的圆括号是否正确配对。

提示：对表达式进行扫描，凡遇 ‘(’ 就入栈，遇 ‘)’ 就将栈顶元素 ‘(’ 出栈；表达式被扫描完毕，栈应为空。

(3) 设计算法把一个十进制整数转换为二至九进制之间的任一进制数输出。

3、实例演练：开关盒布线问题

[问题描述]

假定开关盒为一矩形布线区域，矩形边缘有若干针脚。两个针脚之间通过布设一条金属线路而实现互连，每对互连的针脚被称为网组，连接针脚的金属线被称为电线。如果两条电线发生交叉，则会致使电流短路，所以不允许电线间的交叉。如果网组间可以通过合理安排而不发生电线交叉，我们称该开关盒可以合理布线，或称其为可布线开关盒。图 6.2(a)(b)分别给出了两个开关盒布线的状况，其中(a)为合理布线，(b)为不合理布线。对给定网组和若干个针脚的开关盒，如何判断它能否合理布线？

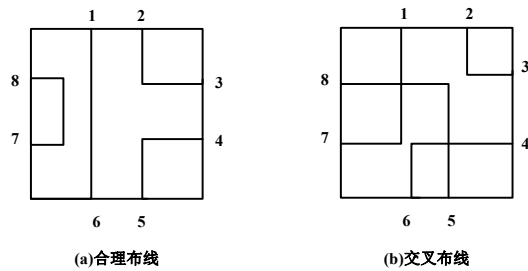


图 6.2 开关盒布线示例图

[测试数据]

有一开关盒如图 6.3 所示，其矩形布线区域的边缘固定有八个针脚，分别以数字 1~8 标识，对应的四个网组分别是(1, 4)、(2, 3)、(5, 6)和(7, 8)。判断该开关盒是否可以产生合理布线？

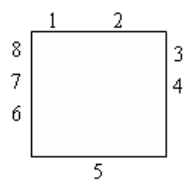


图 6.3 开关盒网组电线分区示意图

[基本要求]

- ① 设计数据结构；
- ② 设计算法完成问题求解；
- ③ 分析算法的时间复杂度和空间复杂度。

[实现提示]

从任意一个针脚开始，沿顺时针或反时针方向来依次访问所有针脚。为判断网组间是否存在电线交叉，我们设置一个栈，将所访问过的针脚依次入栈，若正准备入栈的针脚与栈顶针脚属于同一个网组，则将栈顶针脚出栈。如此反复，直到访问完最后一个针脚。若此时栈空，则表示任意两个网组间都不存在电线交叉，该开关盒可以合理布线；否则，该开关盒不可合理布线。

[思维扩展]

可以使用顺序栈结构实现该问题吗？如果可以，比较它们的时间复杂度和空间复杂度。