

实验三 栈与队列

一 栈与队列的定义

1.1 顺序栈

```
typedef int ElementType;
typedef struct{
    ElementType *elem;
    int stacksize;
    int top;
}SqStack;
```

1.2 或链栈

```
typedef int ElementType;
typedef struct node {
    ElementType data;
    struct node* next;
} Node, *LinkStack;
```

2 队列（链队列）：

```
typedef struct LNode{
    int data;
    struct LNode *next;
}LNode,*LinkList;
```

```
typedef LinkList Queueptr;
typedef struct{
    Queueptr front;
    Queueptr rear;
}LinkQueue;
```

二、具体要求若使用顺序栈,函数定义如下。链栈同理。

1. 编写函数 CreatStack_sq(), DestoryStack_sq(), Push_sq(), Pop_sq(), StackEmpty_sq() 和

StackTraverse_sq(),分别完成创建空栈,销毁栈,入栈,出栈,判断栈是否为空,遍历栈底到栈顶依次打印栈内元素等功能(不要修改原栈),完成后进行测试。

测试要求:在 main 中,建立栈; 判断栈是否为空; 将 0~9 入栈; 将栈顶两个元素出栈,两元素求和后再入栈; 从栈底到栈顶依次打印元素,再从栈顶到栈底打印元素; 销毁栈。

```
void CreatStack_sq(SqStack &S, int msize = STACK_INIT_SIZE){...}  
void DestoryStack_sq(SqStack &S){...}  
void Push_sq(SqStack &S, ElementType e){...}  
bool Pop_sq(SqStack &S, ElementType &e){...}  
bool StackEmpty_sq(SqStack S){...}  
bool StackTraverse_sq(SqStack S){...}
```

2. 编写函数, CreateQueue_L(), DestoryQueue_L(), EnQueue_L(), DeQueue_L(),分别完成创建队列,销毁队列,入队列,出队列等操作,完成后进行测试。

测试要求:在主程序中,建立队列,将 0~9 依次入队列,按入队列顺序出队列并打印,销毁队列。

```
void CreateQueue_L(LinkQueue &Q){...}  
void DestoryQueue_L(LinkQueue &Q){...}  
void EnQueue_L(LinkQueue &Q,int e){...}  
bool DeQueue_L(LinkQueue &Q, int &e){...}
```

3.编写函数 void Knapsack(int w[],int T,int n),完成背包求解问题。

测试数据:

w[6] = {2,8,6,5,1,4};T=11;n=6;

4. (附加题)回文是指正读反读均相同的字符序列,如”abba”和”abdba”均是回文,但”good”不是回文。根据第四章栈和队列所学内容,试写一个算法判定给定的字符向量是否为回文(注意:这里必须利用栈和队列的方式进行判定)。

测试数据:

```
char* ch = “abccba”;  
char* ch = “abccbd”;
```