

实验报告4 参考答案

一、 简答题

1、 栈是一种特殊的线性表，一个栈提供的操作和一个线性表提供的操作是否相同？如果有区别，请说出区别。(10 分)

答：

- 1) 栈提供的操作与线性表提供的操作是有区别的。
- 2) 栈的基本操作是线性表的操作的子集,是操作受控制的线性表。
- 3) 栈仅在表尾进行插入或删除操作,其特点是后进先出。

2、 队列是一种特殊的线性表，一个队列提供的操作和一个线性表提供的操作是否相同？如果有区别，请说出区别。(10 分)

答：

- 1) 队列与线性表的操作是有区别的。
- 2) 队列的操作是线性表操作的子集。
- 3) 队列是一种先进先出的线性表,只允许在表的一端进行插入,而在表的另一端删除元素。

3、 和线性表类似，栈也有两种存储表示方法，顺序栈和链式表示，请解释顺序栈。(10 分)

- 答：顺序栈指的是栈的顺序存储结构是利用一组地址连续的存储单元依次存放自栈底到站顶的数据元素,同时附设指针 `top` 指示元素在顺序栈中的位置。

二、 请按照下面的要求完成实验内容：(20 分)

利用“栈”结构实现“数制转换”问题，对于输入的一个十进制数，能够转换成相应的八进制数，并输入。

由于“栈”结构不是 C 语言提供的，因此需要首先实现栈及其基本操作，才能加以利用解决实际问题。实验步骤如下：

- 1、实现“栈”的结构
- 2、实现“栈”的基本操作
 - (1) 栈初始化操作 `InitStack_Sq`
 - (2) 进栈操作 `Push_Sq`

- (3) 出栈操作 Pop_Sq
- (4) 判断栈是否为空操作 StackEmpty_Sq

3、实现主程序 main

主程序主要是用来控制程序的执行过程,实现数制的转换操作,以及输入、输出。

- 4、程序的编译、链接
- 5、程序的测试

参考答案:

```
#define NULL 0
#define TRUE 1
#define FALSE 0
```

```
#define STACK_INIT_SIZE 100
#define STACKINCREMENT 10
```

```
typedef int SElemType;
```

```
struct Stack{
    SElemType *elem;
    int top;
    int stacksize;
    int incrementsize;
};
```

```
typedef struct Stack *SqStack;
```

```
void InitStack_Sq(SqStack S)
{
    S->elem=(SElemType*)malloc(sizeof(SElemType)*STACK_INIT_SIZE);
    S->top=-1;
    S->stacksize=STACK_INIT_SIZE;
    S->incrementsize=STACKINCREMENT;
}
```

```
void Push_Sq(SqStack S,SElemType e)
{
    if(S->top==S->stacksize)
        printf("Stack is overflow!");
    else
        S->elem[++S->top]=e;
}
```

.....

```

void Pop_Sq(SqStack S,SElemType *e)
{
    if(S->top== -1)
        printf("Stack is Underflow!\n");
    else
        *e=S->elem[S->top--];
}

int StackEmpty_Sq(SqStack S)
{
    if(S->top== -1)
        return(TRUE);
    else
        return(FALSE);
}

main()
{
    SqStack S;
    SElemType *e;
    int N;

    InitStack_Sq(S);

    printf("\n\n");

    printf("Input the Number to be Conversed:");
    scanf("%d",&N);

    while(N){
        Push_Sq(S,N%8);
        N=N/8;
    }

    printf("\n\n");

    printf("the result is:");

    while(!StackEmpty_Sq(S)){
        Pop_Sq(S,e);
        printf("%d",*e);
    }
}

```

.....

