

第二次作业：栈(Stack)

李明（2014*****）

September 28, 2016

请完成以下问题。

问题 1. 设共享栈的结构描述为：

```
#define MAXSIZE 100
#define ERROR 0
#define OK 1

typedef int ElemType;
typedef int Status;
typedef struct {
    ElemType data[MAXSIZE];
    int top1;
    int top2;
} Share_SqStack;
```

请编写能运行的完整程序，自行设计入栈、出栈过程，并输出相关结果。程序应包括以下函数：

```
Status Init(Share_SqStack * S) //初始化共享栈为空
{
    ...
}

Status IsFull(Share_SqStack * S) //判断共享栈是否满栈
{
    ...
}

Status IsEmpty(Share_SqStack * S, int StackNumber) //判断栈1或栈2是否为空
{
    ...
}

Status Push(Share_SqStack * S, ElemType e, int StackNumber)
{
    ...
}

ElemType Pop(Share_SqStack * S, int StackNumber)
{
    ...
}

void PrintS(Share_SqStack * S, int StackNumber) //打印栈1或栈2
{
    ...
}

Status Clear(Share_SqStack * S) //将共享栈清空
{
    ...
}

Status Destroy(Share_SqStack * S) //将共享栈销毁
{
    ...
}

int main(void)
{
    ...
}
```

返回链表的结点个数。

解. 源代码：

```
int length(LinkList L)
{

}
```

运行结果：

```
Initiation:
Stack #1:
Stack #2:
```

```
After pushing 1 2 3 into Stack #1 and 4 5 6 into Stack #2:
Stack #1: 1 2 3
Stack #2: 4 5 6
```

```
After 1 pop in Stack #1 and 2 pop in Stack #2:
Stack #1: 1 2
Stack #2: 4
```

```
After clearing the sharing stack:
Stack #1:
Stack #2:
```

问题 2. 修改数值转换中的 *Convert* 函数，使其能处理十进制数到二、八、十六进制数的转换。最后编写测试函数(即 *main* 函数)，任意输入一个十进制数，输出二、八、十六进制数。提示：可在 *Convert* 函数中设置字符数组

```
char s[] = "0123456789ABCDEF";
```

在输出结果时用

```
putc(s[*e]);
```

解. 源代码：

```
int length(LinkList L)
{

}
```

运行结果：

```
Enter a decimal number: 123
binary: 1111011
octal: 173
hexadecimal: 7B
```