3.2-课堂练习

(1) 建立空顺序表，并初始化；

(2) 按顺序插入元素其值分别为 1-30；

(3) 将其中能被 3 整除的元素删除；

(4) 打印输出所有元素。

注意：调用 ppt 中的相关函数完成上述过程。

```
void InsertList(SeqList *L,int i, ElemType x)
void DeleteList(SeqList *L,int i, ElemType *x)
void InitList(SeqList *L)
int ListLength(SeqList *L)
ElemType GetElem(SeqList *L,int i)
int LocateElem( SeqList *L,     ElemType x)
```

```
SeqList myList;//定义
InitList(&myList);//初始化
int i;
// 按顺序插入元素其值分别为 1-30
for (i=0; i<30; i++)
{ InsertList(&myList,i,i+1);}
// 将其中能被 3 整除的元素删除
```

```
int x=0;
for (i=29; i>0; )
{
If( GetElem(&myList, i) % 3 == 0)
DeleteList (&myList, i, &x);
}
// 打印输出所有元素
for (i=0; i< ListLength(&myList); i++)
{printf("%d\n",GetElem(&myList, i));}
```

思考下述做法有哪些问题？

```
for (int i=0; i<=29; i++)
  if (i%3==0)                    //判断是否能被3整除
    DeleteList (*L, i, *x);      //删除能被3整除的元素
```

注意：

1、 删除操作对于顺序表存储结构的影响

2、 不要利用实际存储值的任何先验知识，程序要具有通用性。

3.3-课堂练习

1

（1）建立空顺序栈并初始化；

（2）入栈 1,2,3,4,5; 出栈两次并输出出栈的元素；最后输出栈中所有的元素。

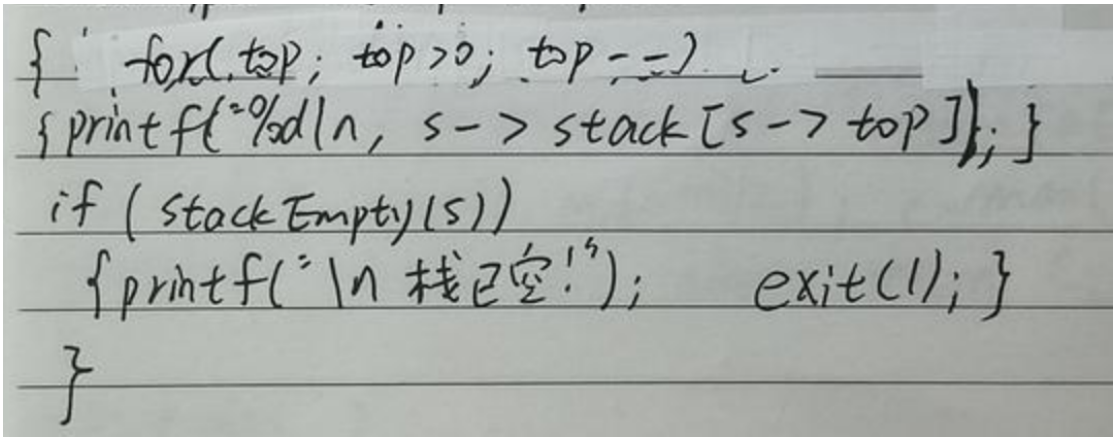
注意：调用 ppt 中的相关函数完成上述过程。

```
void InitStack(SeqStack *s)
int StackEmpty(SeqStack *s)
void push(SeqStack *s,ElemType x)
ElemType pop(SeqStack *s)
ElemType GetTop(SeqStack *s)

    SeqStack myStack;//定义
     InitStack (&myStack);//初始化
int i = 0;
for(i=0; i<5; i++) //入栈 1,2,3,4,5
{ push (&myStack, i+1);}
printf("%d\n", pop(&myStack));
printf("%d\n", pop(&myStack));//出栈两次并输出出栈的元素
//输出栈中所有的元素
int index = myStack->top;
While(index != -1)
{
printf("%d\n", myStack->stack[index]);
index --;
```

}

思考下述做法有哪些问题？



注意：

1、 出栈操作（删除元素）和输出栈中元素（不改变栈中元素）的区别。