

实验四 栈元素排序问题

【对应章节】

栈。

【实验目的】

掌握栈应用的算法设计。

【实验内容】

编写一个程序，按升序对一个字符栈进行排序，即最小元素位于栈顶。最多只能使用一个额外的栈存放临时数据。并输出栈排序过程。

【实验提示】

可采用一个额外的栈 *tmpst* 存放临时数据。设计的功能算法如下。

StackSort (SqStack *&st): 对栈 *st* 中元素排序。其思路是：处理 *st* 栈的栈顶元素 *e*，出栈元素 *e*，将其存放在变量 *tmp* 中。若临时栈 *tmpst* 为空，直接将 *tmp* 进入 *tmpst* 栈；若 *tmpst* 栈不空，将它的元素退栈（放入 *st* 栈中）直到 *tmpst* 栈顶元素小于 *tmp*，再将 *tmp* 进入到 *tmpst* 栈中。进行这样的过程到 *st* 为空，而 *tmpst* 中元素从栈底到栈顶是递增的。再将 *tmpst* 中所有元素退栈并进栈到 *st* 中。

【实验要求】

由用户输入待排序的栈元素，程序运行后，产生类似下图的显示与输出。

```
(1) push the elements 1,3,4,2 onto the stack st.
(2) the process of sorting elements of stack st is:
    st: pop out 2=> tmpst: push 2.
    st: pop out 4=> tmpst: get the top element 2. Since 2<4, exit the loop. tmpst: push 4.
    st: pop out 3=> tmpst: get the top element 4. Since 4>3 tmpst: pop out 4. st: push 4. t
tmpst: get the top element 2. Since 2<3, exit the loop. tmpst: push 3.
    st: pop out 4=> tmpst: get the top element 3. Since 3<4, exit the loop. tmpst: push 4.
    st: pop out 1=> tmpst: get the top element 4. Since 4>1 tmpst: pop out 4. st: push 4. t
tmpst: get the top element 3. Since 3>1 tmpst: pop out 3. st: push 3. tmpst: get the top ele
ment 2. Since 2>1 tmpst: pop out 2. st: push 2. tmpst: push 1.
    st: pop out 2=> tmpst: get the top element 1. Since 1<2, exit the loop. tmpst: push 2.
    st: pop out 3=> tmpst: get the top element 2. Since 2<3, exit the loop. tmpst: push 3.
    st: pop out 4=> tmpst: get the top element 3. Since 3<4, exit the loop. tmpst: push 4.
(3) the sorting of stack st ends.
(4) the popping sequence of st is:1 2 3 4
```