第三章 栈和队列

一、选择题

1. 一个栈的入栈序列是 a,b,c,d,e,则栈的不可能的输出序列是（ ）。

（A） edcba（B）decba（C）dceab （D）abcde

2.栈结构通常采用的两种存储结构是（ ）。

（A） 线性存储结构和链表存储结构（B）散列方式和索引方式

（C）链表存储结构和数组 （D）线性存储结构和非线性存储结构

3.判定一个栈 ST(最多元素为 m0)为空的条件是（ ）。

（A） ST-〉top!=0 （B）ST-〉top==0

（C）ST-〉top!=m0 （D）ST-〉top=m0

4.判定一个栈 ST(最多元素为 m0)为栈满的条件是（ ）。

（A）ST->top!=0 （B）ST->top==0

（C）ST->top!=m0-1（D）ST->top==m0-1

5.一个队列的入列序列是 1,2,3,4,则队列的输出序列是（ ）。

（A）4,3,2,1（B）1,2,3,4（C）1,4,3,2（D）3,2,4,1

6.循环队列用数组 A[0,m-1]存放其元素值,已知其头尾指针分别是 front 和 rear 则当前队列中的元素个数是（ ）

（A）(rear-front+m)%m （B） rear-front+1 （C）rear-front-1（D）rear-front

7.栈和队列的共同点是（ ）

（A） 都是先进后出 （B）都是先进先出

（C）只允许在端点处插入和删除元素（D）没有共同点

8.4 个元素 a1，a2，a3 和 a4 依次通过一个栈，在 a4 进栈前，栈的状态，则不可能的出栈序是（　　）

(A)a4，a3，a2，a1　(B)a3，a2，a4，a1

(C)a3，a1，a4，a2　(D)a3，a4，a2，a1

9.以数组 Q[0..m - 1]存放循环队列中的元素，变量 rear 和 qulen 分别指示循环队列中队尾元素的实际位置和当前队列中元素的个数，队列第一个元素的实际位置是（　　）

(A)rear - qulen　(B)rear - qulen + m

(C)m - qulen　　(D)1 + （rear + m - qulen）% m

二、填空题

1. 栈的特点是_____，队列的特点是_____。

2.线性表、栈和队列都是_____结构,可以在线性表的_____位置插入和删除元素，对于栈只能在_____插入和删除元素，对于队列只能在_____插入元素和_____删除元素。

3.一个栈的输入序列是 12345,则栈有输出序列 12345 是_____。(正确/错误)

4.设栈 S 和队列 Q 的初始状态皆为空，元素 a1，a2，a3，a4，a5 和 a6 依次通过一个栈，一个元素出栈后即进入队列 Q，若 6 个元素出队列的顺序是 a3，a5，a4，a6，a2，a1 则栈 S 至少应该容纳_____个元素。

三、算法设计题

1.假设有两个栈 s1 和 s2 共享一个数组 stack[M],其中一个栈底设在

stack[0]处，另一个栈底设在 stack[M-1]处。试编写对任一栈作进栈和出栈运算的 C 函数 push (x,i)和 pop(i),i=l,2。其中 i=1 表示左边的栈，,i=2 表示右边的栈。要求在整个数组元素都被占用时才产生溢出。

2.利用两个栈 s1,s2 模拟一个队列时,如何用栈的运算来实现该队列的运算？写出模拟队列的插入和删除的 C 函数。

一个栈 s1 用于插入元素,另一个栈 s2 用于删除元素.

参考答案：

一、选择题

1. C 2.A 3. B 4. B 5. B 6.B 7、C 8、C 9、D

二、填空题

1、先进先出；先进后出 2、线性 ； 任何 ；栈顶；队尾；对头 3、正确的 4、3

三、算法设计题

1.

```
#define M 100
elemtype stack[M];
int top1=0,top2=m-1;
int push(elemtype x,int i)
{
if(top1-top2==1) return(1); /*上溢处理*/
else
if(i==1) stack[top1++]=x;
if(i==2)stack[top2--]=x;
return(0);
}
```

```
int pop(elemtype *px,int i)
{
if(i==1)
if(top1==0) return(1);
else
{
top1--;
*px=stack[top1];
return(0);
}
else
if(i==2)
if(top2==M-1) return(1);
else
{
top2++;
*px=stack[top2];
return(0);
}
}
```

2.

```
elemtype s1[MAXSIZE],s2[MAZSIZE];
```

```
int top1,top2;
void enqueue(elemtype x)
{
if(top1==MAXSIZE) return(1);
else
{
push(s1,x);
return(0);
}}
void dequeue(elemtype *px)
{
elemtype x;
top2=0;
while(!empty(s1))
{
pop(s1,&x);
push(s2,x);
}
pop(s2,&x);
while(!empty(s2))
{
pop(s2,&x);
```

```
push(s1,x);

}

}
```