**第3章 栈和队列**

1．选择题

（1）若让元素1，2，3，4，5依次进栈，则出栈次序不可能出现在（ ）种情况。

A．5，4，3，2，1 B．2，1，5，4，3 C．4，3，1，2，5 D．2，3，5，4，1

（2）若已知一个栈的入栈序列是1，2，3，…，n，其输出序列为p1，p2，p3，…，pn，若p1=n，则pi为（ ）。

A．i B．n-i C．n-i+1 D．不确定

（3）数组Ｑ［ｎ］用来表示一个循环队列，ｆ为当前队列头元素的前一位置，ｒ为队尾元素的位置，假定队列中元素的个数小于ｎ，计算队列中元素个数的公式为（ ）。

A．r-f B．(n+f-r)%n C．n+r-f D．（n+r-f)%n

（4）链式栈结点为：(data,link)，top指向栈顶.若想摘除栈顶结点，并将删除结点的值保存到x中,则应执行操作（ ）。

A．x=top->data;top=top->link； B．top=top->link;x=top->link；

C．x=top;top=top->link； D．x=top->link；

（5）设有一个递归算法如下

        int fact(int n) {  //n大于等于0

             if(n<=0) return 1;

             else return n\*fact(n-1);        }

则计算fact(n)需要调用该函数的次数为（ ）。

A． n+1       B． n-1      C． n      D． n+2

（6）栈在 （ ）中有所应用。

A．递归调用 B．函数调用 C．表达式求值 D．前三个选项都有

（7）为解决计算机主机与打印机间速度不匹配问题，通常设一个打印数据缓冲区。主机将要输出的数据依次写入该缓冲区，而打印机则依次从该缓冲区中取出数据。该缓冲区的逻辑结构应该是（ ）。

A．队列 B．栈 C． 线性表 D．有序表

（8）设栈S和队列Q的初始状态为空，元素e1、e2、e3、e4、e5和e6依次进入栈S，一个元素出栈后即进入Q，若6个元素出队的序列是e2、e4、e3、e6、e5和e1，则栈S的容量至少应该是（　）。

A．2 B．3 C．4 D． 6

（9）设计一个判别表达式中左，右括号是否配对出现的算法，采用（　）数据结构最佳。

A．线性表的顺序存储结构 B．队列

C. 线性表的链式存储结构 D. 栈

（10）用链接方式存储的队列，在进行删除运算时（　）。

A. 仅修改头指针 B. 仅修改尾指针

C. 头、尾指针都要修改 D. 头、尾指针可能都要修改

（11）循环队列存储在数组A[0..m]中，则入队时的操作为（　）。

A. rear=rear+1 B. rear=(rear+1)%(m-1)

C. rear=(rear+1)%m D. rear=(rear+1)%(m+1)

（12）最大容量为n的循环队列，队尾指针是rear，队头是front，则队空的条件是（　）。

A. (rear+1)%n==front B. rear==front

C．rear+1==front D. (rear-l)%n==front

（13）栈和队列的共同点是（　）。

A. 都是先进先出 B. 都是先进后出

C. 只允许在端点处插入和删除元素 D. 没有共同点

（14）一个递归算法必须包括（　）。

A. 递归部分 B. 终止条件和递归部分

C. 迭代部分 D. 终止条件和迭代部分

2．算法设计题

（1）将编号为0和1的两个栈存放于一个数组空间V[m]中，栈底分别处于数组的两端。当第0号栈的栈顶指针top[0]等于-1时该栈为空，当第1号栈的栈顶指针top[1]等于m时该栈为空。两个栈均从两端向中间增长。试编写双栈初始化，判断栈空、栈满、进栈和出栈等算法的函数。双栈数据结构的定义如下：

Typedef struct

{int top[2],bot[2]; //栈顶和栈底指针

SElemType \*V; //栈数组

int m; //栈最大可容纳元素个数

}DblStack

（2）回文是指正读反读均相同的字符序列，如“abba”和“abdba”均是回文，但“good”不是回文。试写一个算法判定给定的字符串是否为回文。

（3）设从键盘输入一整数的序列：a1, a2, a3，…，an，试编写算法实现：用栈结构存储输入的整数，当ai≠-1时，将ai进栈；当ai=-1时，输出栈顶整数并出栈。算法应对异常情况（入栈满等）给出相应的信息。

（4）从键盘上输入一个后缀表达式，试编写算法计算表达式的值。规定：逆波兰表达式的长度不超过一行，以$符作为输入结束，操作数之间用空格分隔,操作符只可能有+、-、\*、/四种运算。例如：234 34+2\*$。

（5）假设以I和O分别表示入栈和出栈操作。栈的初态和终态均为空，入栈和出栈的操作序列可表示为仅由I和O组成的序列，称可以操作的序列为合法序列，否则称为非法序列。

①下面所示的序列中哪些是合法的？

A. IOIIOIOO B. IOOIOIIO C. IIIOIOIO D. IIIOOIOO

②通过对①的分析，写出一个算法，判定所给的操作序列是否合法。若合法，返回true，否则返回false（假定被判定的操作序列已存入一维数组中）。

(6）假设以带头结点的循环链表表示队列，并且只设一个指针指向队尾元素站点(注意不设头指针) ，试编写相应的置空队、判队空 、入队和出队等算法。

（7）假设以数组Q[*m*]存放循环队列中的元素, 同时设置一个标志*tag*，以*tag* *==* 0和*tag ==* 1来区别在队头指针(*front*)和队尾指针(*rear*)相等时，队列状态为“空”还是“满”。试编写与此结构相应的插入(*enqueue*)和删除(*dlqueue*)算法。

(8）如果允许在循环队列的两端都可以进行插入和删除操作。要求：

① 写出循环队列的类型定义；

② 写出“从队尾删除”和“从队头插入”的算法。

（9）已知Ackermann函数定义如下:



① 写出计算Ack(m,n)的递归算法，并根据此算法给出出Ack(2,1)的计算过程。

② 写出计算Ack(m,n)的非递归算法。

（10）已知f为单链表的表头指针, 链表中存储的都是整型数据，试写出实现下列运算的递归算法：

① 求链表中的最大整数；

② 求链表的结点个数；

③ 求所有整数的平均值。