表达式求值算法，深刻理解递归算法执行过程中栈的状态变化过程，以更好地使用递归算法。

习题

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（　　）的情况。

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$p_1,p_2,p_3,\cdots,p_n$，若$p_1=n$，则p_i为（　　）。

A. i　　B. $n-i$　　C. $n-i+1$　　D. 不确定

（3）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）链式栈节点为(data,link)，top指向栈顶，若想删除栈顶节点，并将删除节点的值保存到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,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,e1，则栈S的容量至少应该是（　　）。

A. 2　　B. 3　　C. 4　　D. 6

（9）若一个栈以向量V[1..n]存储，初始栈顶指针top设为n+1，则元素x进栈的正确操作是（　　）。

A. top++; V[top]=x;　B. V[top]=x; top++;　C. top--; V[top]=x;　D. V[top]=x; top--;

（10）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

（11）用链接方式存储的队列，在进行删除运算时（　　）。

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填充（见图3.17）。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;                                 //栈最大可容纳的元素个数
}DblStack;
```

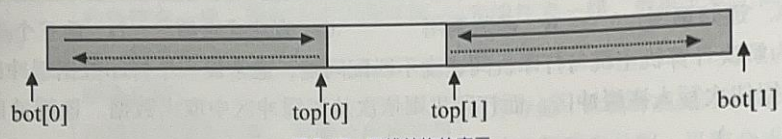

图3.17　双栈结构的表示

（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。试设计算法判定给定的字符序列是否为回文。（提示：将一半字符入栈。）

（3）设从键盘输入一整数的序列$a_1, a_2, a_3, \cdots, a_n$，试设计算法实现：用栈结构存储输入的整数，当$a_i \neq -1$时，将$a_i$进栈；当$a_i = -1$时，输出栈顶整数并出栈。算法应对异常情况（栈满等）给出相应的信息。

（4）从键盘上输入一个后缀表达式，试设计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，输入以“$”作为结束，操作数之间用空格分隔，操作符只可能有“+”“−”“*”“/”4种。例如：234 34 + 2*$。

（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？

P82

教材：

P81： 选择题 1.　（1）、（4）、（8）

P82：算法设计题 2.　（2）