

以下习题供栈和队列章节内容的复习和巩固，不作强制要求，稍后（1-2 周内）会给出参考答案。

栈和队列

习题 A

http://staff.ustc.edu.cn/~leeyi/ds_exer/index3.htm

习题 B （来自数据结构（C 语言版）（第 2 版）章节后习题）

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) 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 ()。

- 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

(12) 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)

(13) 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-1)%n==front

(14) 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

(15) 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