一．是非题

T 8. 栈和队列是操作上受限制的线性表。
F 9. 队列是与线性表完全不同的一种数据结构。
F 10. 队列是一种操作受限的线性表，凡对数据元素的操作仅限一端进行。
T 11. 栈和队列也是线性表。如果需要，可对它们中的任一元素进行操作。
F 12. 栈是限定仅在表头进行插入和表尾进行删除运算的线性表。

二．选择题。

4. 若顺序表中各结点的查找概率不等，则可用如下策略提高顺序查找的效率：若找到指定的结点，将该结点与其后继（若存在）结点交换位置，使得经常被查找的结点逐渐移至表尾。以下为据此策略编写的算法，请选择适当的内容，完成此功能。

顺序表的存储结构为：

```
typedef  struct{
                    ElemType  *elem;  //数据元素存储空间，0 号单元作监视哨
                    int       length;  //表长度
                  }SSTable;
int  search_seq(SSTable  ST,KeyType  key)
{  //在顺序表 ST 中顺序查找关键字等于 key 的数据元素。
   //若找到，则将该元素与其后继交换位置，并返回其在表中的位置，否则为 0。
   ST.elem[0].key=key;
   i=ST.length;
   while(ST.elem[i].key!=key)  ____F____  ;
   if(  ____G____  )
      {ST.elem[i]←→ST.elem[i+1];
            ____E____  ;
      }
   return  i;
}
```

A. i>0　　B. i>=0　　C. i<ST.length　　D. i<=ST.length
E. i++　　F. i--　　G. A 和 C 同时满足　　H. B 和 D 同时满足

C 5. 递归程序可借助于(　　)转化为非递归程序。
a.线性表　　b.队列　　c: 栈　　d.数组

CB 6. 在下列数据结构中(　　)具有先进先出（FIFO）特性，
(　　)具有先进后出(FILO）特性。
a．线性表　　b．栈　　c．队列　　d．广义表

E 7. 若对编号为 1，2，3 的列车车厢依次通过扳道栈进行调度，不能得到（　）的序列。
a:1,2,3　　b:1,3,2　　c:2,1,3　　d:2,3,1　　e:3,1,2　　f:3,2,1

B 8. 在计算递归函数时，如不用递归过程，应借助于( ) 这种数据结构。

A. 线性表　　B. 栈　　C. 队列　　D. 双向队列

C 9. 若带头结点的链表只设尾结点指针。下列选择中（ ）最适用于队列。

A）单链表　B）双向链表　C 循环单链表　D）双向循环链表

C 10. 栈和队列的一个共同点是( )。

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

C 11. 循环队列用数组 A[0..m-1]存放其元素值，设头尾指针分别为 front 和 rear，则当前队列中的元素个数是( )。 长度はm

A. rear-front-1　　B. Rear-front+1

C. (rear-front+m)%m　　D. Rear-front

五．算法设计

7, 顺序栈上实现 POP 算法（需包括栈定义）

8. 设有一个带头结点的单链表 hc，设计一个算法：void split(LinkList *hc, LinkList *&ha, LinkList *&hb,ElemType x,ElemType y)，将 hc 拆分成两个带头结点的单链表 ha 和 hb，其中 ha 的所有结点值均大于等于 x 且小于等于 y，hb 为其他结点。