

## 浙江农林大学继续教育学院考试卷 (A卷)

教学点(分院): 绍兴前进站课程名称: 数据结构 层次: 专升本 学习形式: 函授 考试方式: 闭卷

(2023 学年第一学期)

注意事项: 1、本试卷满分 100 分。

2、考试时间 120 分钟。

题号	一	二	三	四	五	六	七	八	得分
得分									
评阅人									

## 一、选择题 (每题 2 分, 共 24 分)

- 评价一个算法时间性能的主要标准是( )。
  - 算法易于调试
  - 算法易于理解
  - 算法的稳定性和正确性
  - 算法的时间复杂度
- 计算机算法具备有输入、输出、( ) 等五个特性。
  - 可行性、可移植性和可扩充性
  - 可行性、确定性和有穷性
  - 确定性、有穷性和稳定性
  - 易读性、稳定性和安全性
- 带头结点的单链表 head 为空的判定条件是 ( )。
  - head=NULL
  - head->next=NULL
  - head->next==head
  - head!=NULL
- 以下关于线性表的说法不正确的是( )。
  - 线性表中的数据元素可以是数字、字符、记录等不同类型。
  - 线性表中包含的数据元素个数不是任意的。
  - 线性表中的每个结点都有且只有一个直接前趋和直接后继。
  - 存在这样的线性表: 表中各结点都没有直接前趋和直接后继。
- 在顺序表中, 只要知道( ), 就可在相同时间内求出任一结点的存储地址。
  - 基地
  - 结点大小
  - 向量大小
  - 基地址和结点大小
- ( )运算中, 使用顺序表比链表好。
  - 插入
  - 删除
  - 根据序号查找
  - 根据元素值查找
- 一个长度为 n 的顺序表中, 向第 i 个元素之前插入一个新元素时, 需要向后移动 ( ) 个元素。
  - n-i
  - n-i+1
  - n-i-1
  - i

- ( )适合作为经常在首尾两端操作线性表的存储结构。
  - 顺序表
  - 单链表
  - 循环链表
  - 双向链表
- 栈和队列的共同点是 ( )
  - 都是先进后出
  - 都是先进先出
  - 只允许在端点处插入和删除元素
  - 没有共同点
- 一个队列的入列序列是 1 2 3 4, 则队列的输出序列是( )。
  - 4 3 2 1
  - 1 2 3 4
  - 1 4 3 2
  - 3 2 4 1
- 队列与一般的线性表的区别在于( )。
  - 数据元素的类型不同
  - 运算是否受限制
  - 数据元素的个数不同
  - 逻辑结构不同
- “假上溢”现象会出现在( )中。
  - 循环队列
  - 队列
  - 链队列
  - 顺序队列

## 二、填空题 (每空 2 分, 共 28 分)

- 数据的逻辑结构被分为\_\_\_\_\_、线性结构、树形结构和图结构。
- 数据的逻辑结构被分为集合、线性结构、\_\_\_\_\_和图状结构。
- 下面程序段的时间复杂度是\_\_\_\_\_。  

```
i=s=0; while (s<n)
{ i++;s++;}
```
- 树型结构和图形结构合称是\_\_\_\_\_结构。
- 在长度为 n 的顺序存储线性表的第 i 个元素 ( $1 \leq i \leq n$ ) 之前插入一个元素时, 需要向后移动\_\_\_\_\_个元素。
- 在一个长度为 n 的顺序存储的线性表中, 删除第 i 个元素 ( $1 \leq i \leq n$ ) 时, 需要向前移动\_\_\_\_\_个元素。
- 指针 p 指向非空循环单链表 head 的尾结点, 则 p 满足\_\_\_\_\_。
- 已知 L 是带头结点的非空单链表, 且 P 结点既不是第一个数据结点, 也不是最后一个结点, 试从下列提供的答案中选择合适的语句序列, 实现删除 P 结点的直接后继结点的语句序列是\_\_\_\_\_。

- ① P->next = P->next ->next;
- ② P=P->next->next;
- ③ while (P->next!=Q) P=P->next;
- ④ while (P->next->next=Q) P=P->next;
- ⑤ Q=P;
- ⑥ Q=P->next;
- ⑦ P=L;
- ⑧ L=L->next;
- ⑨ free(Q);

9. 在线性结构中，第一个结点\_\_\_\_\_前驱结点，其余每个结点有且只有 1 个前驱结点。

10. 单链表是线性表的\_\_\_\_\_存储表示。

11. 栈是限定仅在\_\_\_\_\_进行插入或删除操作的线性表。

12. 在栈顶指针为 HS 的链栈中，判定栈空的条件是\_\_\_\_\_。

13. 假设以 S 和 X 分别表示进栈和退栈操作，则对输入序列 a、b、c、d、e 进行一系列栈操作 SSXSXSSXXX 之后，得到的输出序列为\_\_\_\_\_。

14. 设栈 S 和队列 Q 的初始状态为空，元素 a、b、c、d、e、f 依次通过栈 S，一个元素出栈后即进入队列 Q。若这 6 个元素出队列的顺序是 b、d、c、f、e、a，则栈 S 的容量至少应该是\_\_\_\_\_。

### 三、算法填空（每空 3 分，共 36 分）

1. 已知一个顺序表中的元素按关键字值非递减有序，下列算法删除顺序表中关键字相同的多余元素，使每个关键字不同的元素在表中只保留一个。

```
void purge_sq(SqlList &la){//删除顺序表 la 中关键字相同的多余元素，即使操作之后的顺序表中只保留操作之前表中所有按关键字值都不相同的元素
```

```
k= -1; //k 指示新表的表尾 for (i=0;i<_____;++i) //顺序考察表中每个元素{j=0;
```

```
while (j<=k &&la.elem[j]!=_____); //在新表中查询是否存在和 la.elem[i]相同的元素
```

```
if (k= -1||j>k) //k= -1 表明当前考察的是第一个元素 la.elem[++k]= la.elem[i];}for
```

```
la.length=_____; //修改表长}purge_sq
```

2. 一个头指针为 head 的单链表，其中每个结点存放一个整数，以下算法将其拆分为两个单链表 head1 和 head2，使 head 1 中仅含有正整数，head 2 中仅含有负整数。

```
void separate(LinkList &head, LinkList &head1,LinkList &head2)
```

```
{//将头指针为 head 的单链表（带头结点）拆分为两个单链表 head1 和 head2,
```

```
//使 head1 中仅含有正整数， head2 中仅含有负整数
```

```
head1=(LinkList)malloc(sizeof(Lnode)); head1->next=NULL;
```

```
head2=_____ ; head2->next=NULL;p=head->next;while(p){ q=p->next;
```

```
if (_____>=0) {p->next=head1->next;head1->next=_____};
```

```
else {p->next=_____;head2->next=p;p=q;} //whilefree(head);}separate
```

3. 设一个长度大于 1 的循环单链表中，既无头结点也无头指针，p 为指向该链表中某个结点的指针，以下为一个删除该结点直接前驱结点的算法。

```
void delete(LinkList p){//在一个既无头结点也无头指针的长度大于 1 的循环链表中， //删除指针 p 所指的某个结点的直接前驱结点 q=_____ ; //查找 p 结点的前驱结点 qwhile(q->next!=p)q=_____ ;r=q; //查找 q 结点的前驱结点 rwhile(_____)r=r->next;r->next=_____ ;free(q);}
```

### 四、计算题（12 分）

1. 设有头指针为 head 的单链表。写算法要求在链表中查找元素值等于 x 的结点，若找到则删除之，如此反复，直至所有值为 x 的元素全部删除为止；若一个也找不到则给出相应提示信息。