## 书面作业1（串之前）

## 作业说明：选择题，填空题在OJ（书面作业1选择填空部分，占比60%。），单击保存，可多次保存。计算题写作业本或纸上，10月9日或16日课堂交。

## 一、选择题（OJ上）

#### 【2分】 1. 以下术语中，从字面上未涉及数据存储结构是（ ）

A.顺序表 B.单链表 C.链队列 D.栈

#### 【2分】 2. 在数据结构概念中，数据的基本单位是（）

A.数据段 B.数据项 C.数据表 D.数据元素

#### 【2分】 3. 在算法设计中，要求算法便于理解和修改是属于算法要求的（）

A.正确性 B.可读性 C.健壮性 D.效率高

#### 【2分】 4. 在数据结构概念中，结构是描述（）

A.数据项的类型 B.数据元素之间的关系 C.数据成员的先后顺序 D.数据对象的取值范围

#### 【2分】 5. 设n为问题规模，以下程序的时间复杂度为（） for (i=1; i<=10000; i++) for (j=1; j<=n; j++) a = a + 1;

A.O(1) B.O(n) C.O(10000n) D.O(n^2)

#### 【3分】 6. 已知顺序表包含100个数据，现在要删除第99号位置的数据，需要移动的数据个数为（）

A.99 B.100 C.1 D.2

#### 【3分】 7. 已知指针p指向单链表L的某个结点，判断p指向的结点是尾结点的条件是（）

A.if (p->next>p) B.if (p->next<p) C.if (p->next==NULL) D.if (p->data==0)

#### 【3分】 8. 已知队列为空，数据1、2、3、4依次逐个进入队列，则出队的数据顺序为（）

A.1234 B.4321 C.1324 D.2413

#### 【3分】 9. 已知栈S为空，数据1、2、3、4依次逐个进入栈S，则栈顶数据为（）

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

#### 【3分】 10. 在设头、尾指针的单链表中，以下()操作与长度n有关。

A.删除第一个结点 B.删除最后一个结点 C.在第一个结点之前插入一个结点 D.在最后一个结点之后插入一个结点

#### 【3分】 11. 设一个链表最常用的操作是在末尾插入节点和删除节点，则选用()最节省时间。

A.单链表 B.单循环链表 C.带尾指针的单循环链表 D.带头结点的双循环链表

#### 【3分】 12. 根据数据元素之间关系的不同，可以把数据结构分为()。

A.动态结构和静态结构 B.内部结构和外部结构 C.逻辑结构和存储结构 D.线性结构和非线性结构

#### 【3分】 13. 设某顺序表中第一个元素的起始存储地址为a,每个元素的长度为b，则第c个元素的起始存储地址是()。（a,b,c均为非负整数）

A.a+b\*c-b B.a+b\*c C.a+b+c D.a+b\*c-c

#### 【3分】 14. 已知一个栈的输出是按ABCD的顺序，可能的入栈顺序有()。

A.ACBD B.ABCD C.DCAB D.BCAD

#### 【3分】 15. 在用不带头结点的单链表存储队列时，其队头指针指向队头结点，其队尾指针指向队尾结点，则在进行删除操作时()。

A.队头、队尾指针都可能修改 B.仅修改队尾指针 C.仅修改队头指针 D.队头、队尾指针都必须要修改

#### 【3分】 16. 栈在()中应用。

A.递归调用 B.表达式计算 C.子程序调用 D.以上都是

#### 【3分】 17. 设p为指向单向循环链表上的某内部结点的指针，则查找p指向结点的直接前驱结点时()。

A.查找时间为O(n) B.查找时间为O(1) C.找不到 D.指针p移动的次数约为n/2

#### 【3分】 18. 带头结点的双循环链表L为空的条件是()。

A.L->next==L&&L->prior==L B.L->next==NULL&&L->prior==L C.L->next==L&&L->prior==NULL D.L->next==NULL&&L->prior==NULL

#### 【3分】 19. 静态链表中的指针表示的是()。

A.下一元素的地址 B.内存储器地址 C.下一个元素在数组中的位置

#### 【3分】 20. 需要分配较大的空间，插入和删除不需要移动元素的线性表，其存储结构为()。

A.单链表 B.静态链表 C.双链表 D.顺序表

#### 【3分】 21. 在双向链表中p所指结点之前插入一个结点q的操作为()。

A.p->prior=q; q->next=p; p->prior->next=q; q->prior=p->prior; B.q->prior=p->prior; p->prior->next= q; q->next= p; p->prior=q->next; C.q->next=p; p->next= q; q->prior->next= q; q->next = p; D.p->prior->next=q; q->next=p; q->prior=p->prior; p->prior=q;

#### 【3分】 22. 对于一个头指针为head的带头结点的单链表，判断为空表的条件是()。

A.head==NULL B.head->next==NULL C.head->next==head D.head!=NULL

#### 【3分】 23. 在双向链表存储结构中，删除p所指的结点时必须修改指针()。

A.p->prior->next = p->next; p->next->prior = p->prior; B.p->prior = p->prior->prior; p->prior->next = p; C.p->next->prior = p; p->next = p->next->prior; D.p->next=p->prior->prior; p->prior=p->next->next;

#### 【3分】 24. 设有两个字符串s和t，求t在s中首次出现的位置的运算称为()。

A.连接 B.求子串 C.求串长 D.模式匹配

#### 【3分】 25. 设目标串为s，模式串为t，在KMP算法中，next[4]=2的含义是()。

A.目标串匹配失败的位置是i=4 B.模式串匹配失败的位置是j=2 C.表示t[4]字符前面最多有两个字符与t开头的两个字符相同 D.表示s[4]字符前面最多有两个字符与t开头的两个字符相同。

## 二、填空题（OJ上）

#### 【3分】 1. 下面程序段的时间复杂度为 \_\_\_： i=1; while(i<=n) i=i+3; （注意时间复杂度表示用大写字母O，纯英文格式）

#### 【3分】 2. 单链表只设头指针，将长度为n的单链表链接在长度为m的单链表之后的算法的时间复杂度为\_\_\_。

#### 【3分】 3. 中缀表达式(A+B)\*(C-D)/(E-F\*G)对应的后缀表达式是\_\_\_。(字符前后，中间不要有空格）

#### 【3分】 4. 判断字符串"([]()[()])"中的括号是否配对时，所需的栈的容量至少是\_\_\_。

#### 【3分】 5. 入栈序列是1,2,3,…n,出栈序列是p1,p2,…,pn，若p2=3，则p3可能取值的个数是\_\_\_个。

#### 【3分】 6. 队列的特点是\_\_\_，栈的特点是\_\_\_

#### 【3分】 7. 设循环队列的空间容量为n，采用损失一个空间的方法，front和rear分别指示队头和队尾位置，则判断队空的条件是\_\_\_,判断队满的条件是\_\_\_，队列中的元素个数是\_\_\_。

#### 【3分】 8. 若用长度为6的数组存储循环队列，且当前rear和front的值分别为1和5，当从队列中删除一个元素，再加入两个元素后，rear和front的值分别为\_\_\_和\_\_\_。

#### 【3分】 9. 根据二元组表示，C=（D，R），其中： D={a，b，c，d，e，f，g，h}， R={<d,b>,<d,g>,<d,a>,<b,c>,<g,e>,<g,h>,<e,f>} 表示\_\_\_结构。

#### 【4分】 10. 设静态链表L(结点元素：data,next)，备用链表头结点为L[0]，数据链表头结点为L[1]。从备用链表头摘一个结点k，插入数据链表p结点之后的语句为:\_\_\_;\_\_\_;\_\_\_;\_\_\_。

## 三、计算题（占比40%）（写在纸上，课堂交）

**1．假定用一维数组a[7]顺序存储一个循环队列，队首和队尾指针分别用front和rear表示，当前队列中已有4个元素：BJ，SH，NJ，WH，其中BJ为队首元素，front的值为3。**

**（1）请画出初始循环队列存储状态。//4分**

**（2）请连续做2次出队运算后的循环队列存储状态；//4分**

**（3）再让GZ，SZ，FS，CS，JM、BH元素依次进队，画出最后的存储状态。若**

**不能入队，说明理由。（采用牺牲一个空间的存储方式）//6分**

**2．假设主串aabaabaabaac，模式串aabaac。**

**（1）求模式串的next值或nextval值（给出详细计算过程，无过程，扣一**

**半）； // 8分**

**（2）写出kmp算法的具体匹配过程。 // 6分，每次失配另起一行**

**3.有4个元素，其入栈次序为A、B、3、\_，在各种可能的出入栈次序中，包含**

**全部四个元素，是合法C语言标识符的有哪几个?**

**给出所有可能的出入栈操作序列及其对应的合法标识符。（用S表示入栈，X**

**表示出栈） // 12分**