《数据结构》期中试卷 （2009 级） 2010-2011 学年第一学期

姓名：_____学号：_____ 成绩：

一、 选择题：（每小题 2 分，共 20 分）

1. 有六个元素 6，5，4，3，2，1 的顺序进栈，下列哪一个不是合法的出栈序列？（ ）
   A. 5 4 3 6 1 2 B. 4 5 3 1 2 6 C. 3 4 6 5 2 1 D. 2 3 4 1 5 6

2. 在一个有 125 个元素的顺序表中插入一个新元素并保持原来顺序不变，平均要移动（ ）个元素。
   A. 8 B. 62.5 C. 62 D. 7

3. 已知广义表 A=((a,b,c),(d,e,f),(h,(i,j)),g)，从 A 表中取出原子项 e 的运算是：（ ）
   A.head(tail(A)) B.head(tail(tail(A)))
   C.head(head(tail(tail(A)))) D.head(tail(head(tail(A))))

4. 循环队列存储在数组 A[0..m]中，设 front 和 rear 分别为队列的头指针和尾指针，则入队时的操作为（ ）。
   A. front=(front +1) mod (m+1) B. rear=(rear+1) mod (m+1)
   C. front=(front +1) mod m D. rear=(rear+1) mod m

5. 在双向循环链表中,在 p 指针所指向的结点前插入一个指针 q 所指向的新结点,其修改指针的操作是( ) (假设双向循环链表的结点结构为(llink,data,rlink)。 A. p->llink=q; q->rlink=p; p->llink->rlink=q; q->llink=q;
   B. p->llink=q; p->llink->rlink=q ; q->rlink= p; q->llink=p->llink;
   C. q->rlink=p; q->llink=p->llink; p->llink->rlink=q; p->llink=q;
   D. q->llink=p->llink; q->rlink=p; p->llink=q; p->llink=q;

6. 一棵完全二叉树上有 1001 个结点，其中叶子结点的个数是（ ）。
   A. 250 B. 500 C. 254 D. 以上答案都不对

7. 已知一棵二叉树的前序遍历结果为 ABCDEF，中序遍历结果为 CBAEDF，则后序遍历的结果为（ ）。
   A. CBEFDA B. FEDCBA C. CBEDFA D. 不定

8. 利用二叉链表存储树时，则根结点的右指针是（ ）。
   A. 指向最左孩子 B. 指向最右孩子 C. 空 D. 非空

9. 设有二维数组 A[0..9, 0..19]，其中每个元素占两个字节，第一个元素的存储地址为 100，若按列优先顺序存储，则元素 A[6,6]存储地址为( )。
   A. 252 B. 132 C. 352 D.232

10. 引入二叉线索树的目的是（ ）
    A. 加快查找结点的前驱或后继的速度
    B. 为了能在二叉树中方便的进行插入与删除
    C. 为了能方便的找到双亲
    D. 使二叉树的遍历结果唯一

二、 填空题：（每小题 2 分，共 20 分）

1. 下面程序段中划线部分的执行次数为_____。

```
int i=0, s=0;
while (++i<=n) {
  int p=1;
  for (int j=1;j<=i;j++) p*=i;
  s=s+p;
}
```

2. 向一个栈顶指针为 H 的链栈中插入一个 s 所指结点时，执行的语句是_____。
3. 如果一棵 Huffman 树 T 有 n_0 个叶子结点， 那么树 T 中共有_____个结点。
4. 在带有一个头结点的链队列 front 中，判定只有一个结点的条件是_____。
5. 对于一个具有 n 个结点的单链表，在已知 p 所指向结点后插入一个新结点的时间复杂度是_____ ；在给定值为 x 的结点后插入一个新结点的时间复杂度是_____。
6. 一棵有 n（n>0）个结点的满二叉树共有_____个叶子和_____非终端结点。
7. 有一个 100*90 的稀疏矩阵（元素类型为整型），非 0 元素有 10 个，设每个整型数占 2 字节，则用三元组表示该矩阵时，所需的字节数是_____。
8. 树的后根遍历序列等同于对该树对应的二叉树进行( )遍历的序列。
9. 具有 256 个结点的完全二叉树的深度为_____。（假设根结点的深度为 0）
10. 循环队列用数组 P 用（0,...,123）共 n 个元素表示，f 为当前队列元素的前一位置，r 为队尾元素的实际位置，当前队列 f 和 r 的值分别为 100 和 32，假定队列中元素个数总小于 124，则队列中元素个数为_____。

三、 判断题：（每题 1 分，共 10 分）

1. （ ）线性表若采用链式存储结构时，占用内存中存储单元的地址一定不连续。
2. （ ）线性表中每个元素都有一个前驱和一个后继。
3. （ ）广义表的长度就是广义表中的原子个数。
4. （ ）任意一棵二叉树中的结点的度都不大于 2。
5. （ ）判断线索二叉树中由 P 所指结点是叶子结点的条件是(P->Lchild==NULL)&&(P->Rchild==NULL)。
6. （ ）采用三元组表方式对稀疏矩阵进行压缩存储时，三元组表中元素个数与矩阵中非零元素个数相同。
7. （ ）队列是一种运算受限的线性表。
8. （ ）二叉树的先序序列中的最后一个结点一定是叶子结点。

9.（ ）完全二叉树中，若一个结点没有左孩子，则它必是叶子结点。

10.（ ）两个栈共享一片连续内存空间时，为提高内存利用率，减少溢出机会，应把两个栈的栈底分别设在这片内存空间的两端。

四、简答题:(每题 5 分，共 20 分)

1. 设二叉树 T 的存储结构如下:

	1	2	3	4	5	6	7	8	9	10
Lchild	0	0	2	3	7	5	8	0	10	1
Data	J	H	F	D	B	A	C	E	G	I
Rchild	0	0	0	9	4	0	0	0	0	0

其中 BT 为树根结点的指针，其值为 Lchild, Rchild 分别为结点的左、右孩子指针域, data 为结点的数据域。

（1）画出二叉树 T 的逻辑结构；

（2）画出二叉树的后序线索树。

2. 用下面数据逐步建成堆。要求画出每加入一个关键码后堆的变化。

（25 11 22 34 5 44 76 61 100 3 14 120）

3. 已知关键字集合 W={ 11, 8, 2, 3, 15, 9 }，以集合中的关键字作为叶子结点的权值而构造哈夫曼树（huffman Tree），画出构造的过程。

4. 设有一字符串 P="3*y-a/y↑2", 试写出利用栈将 P 改为"3y*ay2↑/-"的操作步骤。(请用 X 代表扫描该字符串过程中顺序取一字符进栈的操作，用 S 代表从栈中取出一字符加入到新字符串尾的出栈操作。例如，要使"ABC"变为"BCA"，则操作步骤为 XXSXSS)。

五、 算法设计题:（每题 10 分，共 30 分）

1.已知 L 是带表头结点的单链表(表中元素个数 >= 2)，P 指向某结点（非第一结点），删除 P 结点的直接前驱语句是:

2. 下面的算法是将整型数组 A[0..n-1]中的元素划分为两部分，使得左边的所有元素均为奇数，右边的所有元素均为偶数，补充完成 A,B,C,D 四个空（每处空并非仅有一条语句）:

```
void  Partition(int A[ ])
{  i = 0; j = n-1;
   while  ( ____A____ )
   {
        while (i < j && ____B____ )  i++;
        while (i < j && ____C____ )  j--;
        if (i < j)  ____D____ ;
   }
}
```

3. 二叉树以二叉链表的方式存储，设计算法输出二叉树中所有的叶子结点，同时给出每个叶子结点到根结点的路径的长度。

一

1. C
2. B
3. D
4. B
5. C
6. D
7. A
8. C
9. D
10. A

二

1. n(n+1)/2
2. S->link = H; H= S;
3. $2n_0$-1
4. front->link->link==NULL; 或 front->link->link==rear;
5. O(1);O(n)
6. (n+1)/2;(n-1)/2
7. 60
8. 中序
9. 8
10. 56

三

1. F
2. F
3. F
4. T
5. F
6. T
7. T
8. T
9. T
10. T

四

3

1.

解法 1）

```
WPL(T:BNode *):int;
 { n= 0;   WPL1(T,0);
          WPL= n
 };
void   WPL1(T:BNode *; h:int);
 {
    if ( T !=NULL )
      if ( (T->Lchild==NULL) && (T->Rchild==NULL)
        )   n= n+T->data*h
        ; else { WPL1(T->Lchild,h+1);   WPL1(T->Rchild,h+1)   }
 };
```

解法 2）

```
WPL(T:BNode *):int;
 {
  if ( T ==NULL
  )  WPL= 0
  ; else if ( (T->Lchild==NULL) && (T->Rchild==NULL)
        )  WPL= 0
        ; else WPL= T->data+WPL(T->Lchild+WPL(T->Rchild)
};
```

数据结构期中考试试题

（2010211124~29）

一.　　单项选择题（总计 20 分，2 分/题）

1. 在线性表中最常用的操作是在最后一个元素之后插入一个元素和删除第一个元素，则采用（　　）存储方式最节省运算时间。

   A.单链表　　B.仅有头指针的单循环链表

   C.双链表　　D.仅有尾指针的单循环链表

2. 链表不具有的特点是（　　）。

   A.可随机访问任一元素　　B.插入、删除不需要移动元素

   C.不必事先估计存储空间　　D.所需空间与线性表长度成正比

3. 一个栈的输入序列为 12345，则下列序列中是栈的输出序列的是（）。

   A.23415　　B.54132　　C.31245　　D.14253

4. 设循环队列中数组的下标范围是 1~n，其头尾指针分别为 f 和 r，则其元素个数为（）。

   A.r-f　　B.r-f+1　　C.(r-f) mod n +1　　D.(r-f+n) mod n

5. 数组 A[1..5,1..6]的每个元素占 5 个单元，将其按行优先顺序存储在起始地址为 1000 的连续的内存单元中，则元素 A[5,5]的地址为（）。

   A.1140　　B.1145　　C.1120　　D.1125

6. 数据结构被形式地定义为（D，R），其中 D 是①（）的有限集合，R 是 D 上的②（）有限集合。

   ①A．算法　　B．数据元素　　C．数据操作　　D．逻辑结构

   ②A．操作　　B．映象　　C．存储　　D．关系

7. 在单链表中 p 元素后面插入 q 指针所指的新元素时应进行的操作是(　　)。

   A. p->next = q, q->next = p->next　　B. q->next = p->next, p->next = q

   C. p ->next->next = q, q->next = p　　D. q->next = p, p->next->next = q

8. 在下面这段代码中，假定赋值运算为主要操作，那么它的时间复杂度为（）。

```
for(I=0; I<n; I++)
    for(J=I; J<n; J++) {X[I * n +J]= 0;}
```

   A．O(n)　　B. O(2n)　　C. $O(n^2)$　　D. $O(n\log^n)$

9. 不带头结点的单链表 head 为空的判定条件是(　　)

   A. head=NULL

   B. head - >next=NULL

   C. head- >next=head

   D. head!=NULL

10. 一个栈的入栈序列为 a,b,c,d,e，那么不可能出现输出序列为(　　)

   A. edcba　B　decba　　C deeab　D abcde

二.　　判断题（总计 15 分，1 分/题）

1. （×）串长度是指串中不同字符的个数。
2. （×　）数组可以看成是线性结构的一种推广，因此可以对它进行插入、删除等运算。
3. （×　）在顺序表中取出第 i 个元素所花费的时间与 i 成正比。
4. （√　）在栈满情况下不能作进栈操作，否则产生“上溢”。
5. （×　）线性表的长度是线性表所占用的存储空间的大小。
6. （×）双循环链表中，任意一结点的后继指针均指向其逻辑后继。
7. （×）在对链队列做出队操作时，不会改变 front 指针的值。
8. （×）如果两个串含有相同的字符，则说它们相等。
9. （√　）若一个栈的输入序列为 123…n，其输出序列的第一个元素为 n，则其输出序列的每个元素 a_i 一定满足 a_i =n-i+1。(i=1,2,...,n)
10. （×）线性表就是顺序表。
11. （×）设指针 p 指向单链表中的一个结点，则语句序列 u=p->next; u=u->next 将删除一个结点。
12. （×）链表的每个节点都恰好有一个指针。
13. （×）稀疏矩阵压缩存储后丧失随机存取功能。(题目不严格)
14. （×）链式存储结构中一个结点的空间可以不连续。
15. （×)长度为 m 的单链表连接在长度为 n 的单链表之后的算法的时间复杂度为 O(n)。

三.　　填空题（2 分/空，共计 20 分）

1. 在双循环链表 L 中，指针 p 所指结点为尾结点的条件是（p->next=L)。
2. 在单链表中，删除指针 p 所指结点的后继结点的语句是（p->next=p->next->next，free(p->next);)。
3. 若某串的长度小于一个常数，则采用（定长顺序）存储方式最节省空间。
4. 设循环队列的容量为 30,设队头 f 指向队列中的第一个元素，队尾 r 指向队列中的最后一个元素，　若 f=24 且 r=10,则队列中有（　16　）个元素。
5. 已知栈的输入序列为 1,2,3,...,n，输出序列为 $a_1, a_2, a_3,..., a_n$，符合 a_2=n 的输出序列共有（ n-1　）种。
6. 已知数组 A[1..10,1..10]为对称矩阵，其中每个元素占 5 个单元。现将其下三角部分按行优先次序存储在起始地址为 1000 的连续内存单元中，则元素 A[5，6]对应的地址为（(4*10+6-1）*5+1000=1225)。
7. 对于一个具有 n 个结点的单链表，在已知 p 所指向结点后插入一个新结点的时间复杂度是 O(1)　在给定值为 x 的结点后插入一个新结点的时间复杂度是　（ O(n)　）。
8. 线性表的元素长度为 4，LOC(a1)=1000，　则 LOC(a10)=（ 1036 (题目不严格)）。
9. 在一个带头结点的单向循环链表中，p 指向尾结点的直接前驱，则指向头结点的指针 head 可用 p 表示为(　head=p->next->next　）。

10. 在 KMP 算法中，模式串’aababaaaba’的 next 数组值为（0121212334）.

四. 问答题（共计 15 分）

1、 利用两个栈 S1S2 模拟一个队列时，如何用栈的运算来实现队列的运算（插入一个元素、删除一个元素、判断队列为空）。请阐述实现思路，并用 C 语言描述实现方法。

```
Enqueue(Element e)
{
    If (s1.length == maxsize) return OVERFLOW;
    Push(e);
}
Dequeue(Element *e)
{
 If (s1.isEmpty())      return OVERFLOW;
While (!s1.isEmpty())   Push(s2,pop(s1,e));
Pop(s2,e);
While(!s2.isEmpty())   Push(s1,pop(s2,e));
}
isEmpty(){
    return s1.isEmpty();
}
```

2、 设目标串为 t= ‘abcaabbabcabaacbacba’，模式 p= ‘abcabaa’。

计算模式 p 的 next 函数值 ，试判断第几趟匹配成功，写出匹配过程。

Next={0,1,1,1,2,3,2}

第一趟 abcaabbabcabaacbacba

abcabaa

第二趟 abcaabbabcabaacbacba

abcabaa

第三趟 abcaabbabcabaacbacba

abcabaa

第四趟 abcaabbabcabaacbacba

abcabaa

第五趟 abcaabbabcabaacbacba

abcabaa

第六趟 abcaabbabcabaacbacba

abcabaa

3、 已知三维数组 A[1..6,0..7, -1..4]，每个元素占用 4 个字节，存储器按字节编址。已知 A 的起始存储位置是 1024，计算

1）数组 A 占用的存储空间大小（1 分）

2）按低下标优先存储时，A[3,5,2]的第一个字节的地址（2 分）

类似行优先

3）按高下标优先存储时，A[3,5,2]的第一个字节的地址（2 分）

类似列优先

1）[(6-1+1) ×(7-0+1) ×(4-(-1)+1)]× 4
=[6 × 8 × 6] × 4=1152 字节

2）1024+[(3-1)×8×6+(5-0)×6+(2-(-1))]×4=1540

3）1024+[(3-1)+(5-0)×6+(2-(-1))×8×6]×4=1728

五. 算法设计题（10 分/题，总计 30 分）

1. 单链表中，每个结点含有 5 个正整型的数据元素（若最后一个结点的数据元素不满 5 个用 0 填充），试编写一算法查找值为 n(n>0)的数据元素所在的结点指针以及在该结点中的序号，若链表中不存在该数据元素，则返回空指针。

设每个节点中 5 个数据元素保存在 data[5]中，且设 p 指向第一节点

```
Node *findvalue(int n, int *index) {
  While(p!=null) {
    for (i=0;i<5;i++)
     if (p->data[i] == n) {
        *index = i;
         return p;
      }
    p=p->next;
  }
  *index = 0;
  Return NULL;
}
```

2. 假设在长度大于 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， 编写一个函数删除该结点的前驱结点。


```
pre = p;
q = p->next;
While(q->next != p)   {pre = q; q = q->next;}
pre->next = p;
free(q);
```

3. 已知两个整数集合 A 和 B， 它们的元素分别依元素值递增有序存放在两个单链表 HA 和 HB 中。下述算法的功能是什么?

```
void unn(ha,hb,hc)
node *ha,*hb,*hc;
{
node *p,*q,*r,*s;
hc={node *}malloc(sizeof(node));
/*建立一个头结点， r 总是指向 HC 链表的最后一个结点*/
r=hc;p=ha;q=hb;
while(p!=NULL&&q!=NULL)
{
  if (p->data<q->data)
    {
      s=(node*)malloc(sizeof(node));
      s->data=p->data;
      r->next=s;
      r=s;
      p=p->next;
    }
 else if (p->data>q->data)
    {
      s=(node*)malloc(sizeof(node));
      s->data=q->data;
      r->next=s;
      r=s;
      q=q->next;
    }
```

```
    else
        {
         s={node*}malloc(sizeof(node));
        q=q->next;
        }
}
if(p= =NULL)
while (q!=NULL)
{
    s=(node * )malloc(sizeof(node));
    s->data=q->data;
    r->next=s;
    r=s;
    q=q->next;
}
if (q= =NULL)
while(p!=NULL)
{
     s=(node*)malloc(sizeof(node));
        s->data=p->data;
        r->next=s;
        r=s;
        p=p->next;
}
r->next=NULL;
s=hc;
hc=hc->next;
free(s);
}
```