**华中师范大学信息管理系2016-2017学年**

系别： **信息管理系** 专业： 信管、电商 年级： 2015  学生姓名： 学号：

--------------------------------------------------------------------------------------------------------------- ---------------------------------------------------------

**第一学期期中考试试卷**

**课程名称：数据结构**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **题型** | **选择** | **填空** | **简述** | **完成算法** | **编程** | **总分** |
| **分值** | **20** | **24** | **12** | **32** | **12** | **100** |
| **得分** |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| **得分** |  | **一、选择题：（共10题，每题2分，共20分）** |
| **评阅人** |  |

1.下列有关线性表的叙述中，正确的是（　A　）

A 线性表中的元素之间是线性关系

B 线性表中至少有一个元素

C 线性表中任何一个元素有且仅有一个直接前趋

D 线性表中任何一个元素有且仅有一个直接后继

2. 若让元素1，2，3依次进栈，则出栈次序不可能是（ C ）。

A. 3，2，1　 B. 2，1，3 C. 3，1，2 D. 1，3，2

3. 栈结构通常采用的是（ A ）。

A 顺序存贮结构和链式存储结构

B 散列方式和索引方式

C 链表存储结构和数组

D 线性存储结构和非线性存储结构

4. 二维数组M的元素是4个字符（每个字符占一个存储单元）组成的串，行下标i的范围从0到4，列下表j的范围从0到5，M按行存储时元素M[3][5]的起始地址与M按列存储时元素（ B ）的起始地址相同。

A M[2][4]

B M[3][4]

C M[3][5]

D M[4][4]

5. 在一个单链表中，若删除p所指结点的后续节点，则执行（ A ）

A p->next=p->next->next; B p=p->next;p->next=p->next->next

C p->next=p->next; D p=p->next->next;

6. 不带头结点的单链表head为空的判定条件是（ A ），带头结点的单链表head为空的判定条件是（ B ）。

A head=NULL B head->next=NULL

C head->next=next D head!=NULL

7. 某二叉树的前序遍历结点访问顺序是abdgcefg，中序遍历的结点访问顺序是dgbaechf，则其后序遍历的结点访问顺序是（ D ）。

   A bdgcefha     B gdbecfha    C bdgaechf    D gdbehfca

8. 下列对于线性表的叙述中错误的是（ D ）

A 若用顺序存储，表中元素的存储位置是连在一起的

B 若用链表存储，便于插入和删除操作

C 若用链表存储，不需要占用一片相邻的存储空间

D 表的插入和删除只允许在表的一端进行

9．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（ C ）。

A H->next=s; B s->next= H->next; H->next=s;

C s->next=H; H=s; D s->next=H; H=H->next;

10．设矩阵A是一个对称矩阵，为了节省存储，将其下三角部分按行序存

放在一维数组B[1,n(n-1)/2]中，对下三角部分中任一元素ai,j(i>=j)，在

一维数组B的下标位置k的值是（ B ）。

A. i(i-1)/2+j-1 B. i(i-1)/2+j

C. i(i+1)/2+j-1 D. i(i+1)/2+j

|  |  |  |
| --- | --- | --- |
| **得分** |  | **二、填空题：（共12空，每空2分，共24分）** |
| **评阅人** |  |

1.数据结构被形式地定义为（K,R），其中K是 数据元素 的有限集合，R是K上的 逻辑结构 的有限集合。

2.二维数组A[10..20,5..10]，采用行序为主存储方式，每个元素占4个存储单元，并且A[10][5]的存储地址是1000，则A[18][9]的地址是 1208 。

3.链表、栈和队列都是 线性 结构，可以在链表的 任何 位置插入和删除元素；对于栈只能在 栈尾 插入和删除元素；对于队列只能够在 队尾 插入元素和 队首 删除元素。

4. 在一个链式队列H中，判定只有一个结点的条件是 H->front = H->rear 。

5. 循环链表中最后一个结点的 指针域 指向头结点，整个链表形成一个 环 。

6. 设矩阵A是一个对称矩阵，为了节省存储，将其下三角部分按行序存储在一维数组B[]中，对下三角部分中任一元素a(i,j) (i>=j)，在一维数组B中下标k的值是 i(i-1)/2+j 。

|  |  |  |
| --- | --- | --- |
| **得分** |  | **三、简述题：（共2题，每题6分，共12分）** |
| **评阅人** |  |

1. 给出(普通)双向链表中结点的结构描述（用C语言，没有表头），并简述单链表、单向循环链表、(普通)双向链表以及双向循环链表的区别。

typedef struct DuNode{

ElemType data;

struct DuNode \*prior;

struct Dunode \*next;

}DuLNode, \*DuLinkList;

单链表有一个数据域和一个指针域，表尾节点的指针域置NULL；单向循环链表有一个数据域和一个指针域，表尾节点的指针域指向第一个节点；双向链有一个数据域和两个指针域LEFT和RIGHT，表第一个节点的LEFT域置为NULL，最后一个结点的RIGHT域置为NULL；双向循环链表有一个数据域和两个指针域LEFT和RIGHT，表尾结点的RIGHT域指向第一个结点，第一个结点的LEFT域指向最后一个结点。

2．试举一例，说明对相同的逻辑结构，同一种运算在不同的存储方式下实现，其运算的效率不同，并比较两种存储方式在不同运算下的效率。

线性表中的插入、删除操作，在顺序存储方式下平均移动近一半的元素，时间复杂度为O(n)；而在链式存储方式下，插入和删除时间复杂度都是O(1)。

|  |  |  |
| --- | --- | --- |
| **得分** |  | **四、阅读算法题：（共1题，每题6分，共6分）** |
| **评阅人** |  |

LinkList mynote(LinkList &L,a[])

{ //L是不带头结点的单链表的头指针

if(L&&L->next) q = L;

S1： while(q->next) q=q->next;

for(int i=0, i<=2; i++){

LinkList p = (Linklist)malloc(sizeof(LNode));

S2： p->data = a[i]; q->next = p; q=p;

}

p->next =NULL;

}

}

请回答下列问题：

⑴说明语句S1的功能；

⑵说明语句组S2的功能；

⑶设LinkList是(3,2,4,6,7）,a[]={5,0,1,8,9},写出算法执行后的返回值所表示的线性表。

（1）创建新节点

（2）将a[]内元素依次插入到链表尾部

（3）返回的线性表为（3,2,4,6,7,5,0,1）

|  |  |  |
| --- | --- | --- |
| **得分** |  | **五、完成算法：（共9空，每空2分，共18分）** |
| **评阅人** |  |

1．下面是带头指着head的单链表的就地逆置，即利用原带头结点单链表head的节点空间把数据元素(a0,a1,...am-1)逆置为(am-1,...,a1,a0)。

void Converse\_L(LinkList &head)

{

LinkList p, q;

p = head->next;

head->next = NULL; //将表头砍断

while(p)

{

q=p; //利用指针p, q将链表逆置

p = p->next;

q->next = head->next;

head->next=q;

}

}

2．利用堆栈完成数制转换问题

void TransForm(long N, int d)

//N表示需要转换的10进制数，d为需要转换成的数制

LinkStack S;

int x;

InitStack\_(S);

while(N)

{

Push\_L(S, N%d);

N = N/d;

}

while( !StackEmpty\_L(s) )

{

Pop\_L(S,x);

cout<<x;

}

cout<<endl;

}//Transform

|  |  |  |
| --- | --- | --- |
| **得分** |  | **六、编程题：（共2题，每题10分，共20分）** |
| **评阅人** |  |

1．设计以单链表存储的两个集合求交集的算法

void Interaction(LinkList La,LinkList Lb,LinkList &Lc)

{

LinkList pa,pb,pc;

LinkListSort(La);

LinkListSort(Lb); //调用第6题的排序函数

pa=La->next;

pb=Lb->next;

Lc=(LinkList)malloc(sizeof(LNode));

Lc->next=NULL;

while(pa && pb)

if(pa->data < pb->data)

pa = pa->next;

else if(pa->data>pb->data)

pb=pb->next;

else

{

pc=(LinkList)malloc(sizeof(LNode));

pc->data=pa->data;

pc->next=Lc->next;

Lc->next=pc;

pa=pa->next;

pb=pb->next;

}

}

或者

LinkList pa,pb,pc;

InitList(c);

pa=La->next; pb=Lb->next;

Lc=(LinkList)malloc(sizeof(LNode)); Lc->next=NULL;

while(pa)

{while(pb)

{if(p->data==q->data)

{pc=(LinkList)malloc(sizeof(LNode)); pc->data=pb->data; }

pb=pb->next;

}

pa=pa->next;

}

去重。

2．设计以下算法：

(1)往顺序表L的第i个元素（0≤i≤L.len）之后插入一个新元素x。

(2)从顺序表L中删除其值等于d的所有元素。

(3)将两个有序顺序表A和B合并成一个有序表C。

注：为准确起见，列出算法第一行

(1) bool Sq\_Insert(SqList L, int i, ElemType x)

(2) bool Sq\_delete(SqList &L, int d, ElemType &e)

(3) bool Sq\_merge(SqList A, SqList B, SqList &C)

(1) 解答: int j;

if(i>0){for (j=L.len ; j>= i＋1;--j )

L.elem[j＋1]＝L.elem[j]; return true;

L.elem[i＋1]＝x;

L.len＝len＋1; } else return false;

(2) 解答:

int i＝1,j ;

while (i<＝L.len )

{

if (L.elem[i]= =x ){

for( j＝i＋1 ;j<= L.len ;++j)

{L.elem[j-1]=L.elem[j];

L.len--;

return true;

}

else return false;}

(3)解答：bool Merge(SeqList A, SeqList B, SeqList &C){

if(A.length+B.length>C.maxSize) return false;

int i=0,j=0,k=0;

while(i<A.length && j<B.length)//或者for

{

if(A.data[i] < B.data[j])

C.data[k] = A.data[i]; i++; k++;

else

C.data[k]=B.data[j]; j++; k++

}

while(i<A.length) //A没比较完

C.data[k++] = A.data[i++];

while(j<B.length) //B没比较完

C.data[k++] = B.data[j++];

C.length=k;

return true;

}