51Ch2 习题参考答案

16.循环链表的主要优点是(D)；(A)不再需要头指针了；(B)已知某个结点的位置后，能够容易找到他的直接；(C)在进行插入、删除运算时，能更好的保证链表不；(D)从表中的任意结点出发都能扫描到整个链表；17.下面关于线性表的叙述错误的是(B)；(A)线性表采用顺序存储，必须占用一片地址连续的；(B)线性表采用顺序存储，便于进行插入和删除操作；(C)线性表采

16. 循环链表的主要优点是(D) 。

(A)不再需要头指针了

(B)已知某个结点的位置后，能够容易找到他的直接前趋

(C)在进行插入、删除运算时，能更好的保证链表不断开

(D)从表中的任意结点出发都能扫描到整个链表

17. 下面关于线性表的叙述错误的是(B)。

(A) 线性表采用顺序存储，必须占用一片地址连续的单元；

(B) 线性表采用顺序存储，便于进行插入和删除操作；

(C) 线性表采用链式存储，不必占用一片地址连续的单元；

(D) 线性表采用链式存储，便于进行插入和删除操作；

18. 单链表中，增加一个头结点的目的是为了（C ）。

(A) 使单链表至少有一个结点 (B)标识表结点中首结点的位置

（C）方便运算的实现 (D) 说明单链表是线性表的链式存储

19. 若某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（D ）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

20. 若某线性表中最常用的操作是取第 i 个元素和找第 i 个元素的前趋元素，则采用（ ）存储方式最节省运算时间（B ）。

(A) 单链表 (B) 顺序表

(C) 双链表 (D) 单循环链表

21. 一个向量(一种顺序表)第一个元素的存储地址是 100，每个元素的长度为 2，则第 5 个元素的地址是_____。

A. 110 B. 108

C. 100 D. 120

答：B

[第 5 个元素的地址=100+2*(5-1)=108]

22. 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;

答：A

23. 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;

答：B

24. 在循环双链表的 p 所指结点之后插入 s 所指结点的操作是_____。

A. p->right=s; s->left=p; p->right->left=s; s=->right=p->right;

B. p->right=s; p->right->left=s; s->left=p; s->right=p->right;

C. s->left=p; s->right= p->right; p->right=s; p->right->left=s;

D. s->left=p; s->right=p->right; p->right->left=s; p->right=s;

答：D

25. 在一个单链表中，已知 q 所指结点是 p 所指结点的前驱结点，若在 q 和 p 之间插入 s 结点，则执行_____。

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

C. q->next=s; s->next=p; D. p->next=s; s->next=q;

答：C

26. 从一个具有 n 个结点的单链表中查找其值等于 x 结点时，在查找成功的情况下，需平均比较_____个结点。(参见网上讲义：1.4.2 例 1_5)

A. n; B. n/2;

C. (n-1)/2; D. (n+1)/2;

答：D

27. 给定有 n 个结点的向量，建立一个有序单链表的时间复杂度_____。

A. O(1); B. O(n);

C. O(n2); D. O(nlog2n);

答：C

三、 填空题

28. 在一个长度为 n 的向量中的第 i 个元素(1≤i≤n)之前插入一个元素时，需向后移动_____个元素。

答：n-i+1

29. 在一个长度为 n 的向量中删除第 i 个元素(1≤i≤n)时，需向前移动_____个元素。

答：n-i

30.在一个单链表中 p 所指结点之前插入一个由指针 s 所指结点,可执行以下操作：

s->next=__p->next_____;

p->next=s;

t=p->data;

p->data=___s->data_______;

s->data=___t_______;

四、算法设计题：

31. 有一个单链表（不同结点的数据域值可能相同），其头指针为head，编写一个函数计算数据域为 x 的结点个数。

解：本题是遍历通过该链表的每个结点，每遇到一个结点，结点个数加 1，结点个数存储在变量 n 中。实现本题功能的函数如下：

```
int count (head, x)

node *head;

ElemType x;

{

/*本题中 head 为链头指针，不含头结点*/

node *p;

int n=0;

p=head;

while (p!=NULL)

{

if (p->data= =x) n++;

p=p->next;

}

return(n);

}
```

32. 有一个有序单链表（从小到大排序），表头指针为 head，编写一个函数向该单链表中插入一个元素为 x 的结点，使插入后该链表仍然有序。

解：本题算法的思想是先建立一个待插入的结点，然后依次与链表中的各结点的数据域比较大小，找出该结点的位置，最后插入该结点。实现本题功能的函数如下：

```
node *insertorder(head, x)

node *head;
```

```
ElemType x;

{

/*本题中 head 为链头指针，不含头结点*/

node *s, *p, *q;

s=(node *)malloc(sizeof(node)); /* 建立一个待插入的结点*/

s->data=x;

s->next=NULL;

if (head==NULL ‖ x<head->data) /* 若单链表为空或 x 小于第一个结点的 data 域*/

{

s->next=head; /* 则把 s 结点插入到表头后面*/

head=s;

}

else

{

q=head;

/*为 s 结点寻找插入位置,p 指向待比较的结点,q 指向 p 的前驱结点*/

p=q->next;

while (p!=NULL && x>p->data) /* 若 x 小于 p 所指向的 data 域值*/ if (x > p->data)

{

q=p;

p=p->next;

}

s->next = p; /* 将 s 结点插入到 q 和 p 之间*/
```

```
q->next=s;
}
return(head);
}
```

33. 编写一个函数将一个头指针为 a 的单链表 A 分解成两个单链表 A 和 B，其头指针分别为 a 和 b，使得 A 链表中含有原链表 A 中序号为奇数的元素，而 B

链表中含有原链表 A 中序号为偶数的元素，且保持原来的相对顺序。

解：其函数是将单链表 A 中的所有偶数序号的结点删除，并在删除时把这些结点链接起来构成单链表 B。实现本题功能的函数如下：

```
void disa(a, b)
node *a, *b;
{
/*本题中 a、b 为链头指针，不含头结点*/
node *r, *p, *q;
p=a;
b=a->next;
r=b;
while (p!=NULL && p->next!=NULL)
{
q = p->next; /* q 指向偶数序号的结点*/
p->next=q->next; /* 将 q 从原 A 中删除掉*/
r->next=q; /* 将 q 结点链接到 B 链表的末尾*/
r=q /* r 总是指向 B 链表的最后一个结点*/
p=p->next; /*p 指向原链表 A 中的奇数序号的结点*/
}
```

r->next=NULL; /* 将生成 B 链表中的最后一个结点的 next 域置空*/ }

34. 假设有两个已排序的单链表 A 和 B，编写一个函数将它们合并成一个链表 C 而不改变其排序性。

解：这里采用链表合并的方法，设原两链表的头指针分别为 p 和 q，每次比较 p->data 和 q->data 的值，把值较小的结点作为新链表的结点，这一过程直到一个链表为空为止。当一个链表为空而另一个链表不为空时，只要将不空的链表指针赋给新链表中最后一个结点的指针即可。实现本题功能的函数如下：

```
node *mergelink(p, q)

node *p, *q;

{

/*本题中 p、q 为链头指针，不含头结点。*/

/*但为操作方便，过程中要为链表 C 建立一个临时头结点。*/

node *h, *r;

h=(node *)malloc(sizeof(node)); /* 建立头结点*/

h->next=NULL;

r=h;

while (p!=NULL && q!=NULL)

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

{

r->next=p;

r=p;

p=p->next;

}

else

{
```

```
r->next=q;

r=q;

q=q->next;

}

if (p= =NULL)

/* 若 A 链表的结点已取完，则把 B 的所有余下的结点链接 C 中*/ r->next=q;

if (q= =NULL)

/*若 A 链表的结点已取完，则把 B 的所有余下的结点链接 C 中*/ r->next=p;

/*下面三句要去掉链表 C 的头结点，如果不想去掉，则不要这三句*/

p=h->next;

h=h->next;

free(p);

return h;

}
```

35. 设 A=（a1,…,am）和 B=（b1,…,bn）均为顺序表，A’和 B’分别为 A 和 B 中

除去最大共同前缀后的子表（例如，A=（x，y，y，z，x，z），B=（x，y，y，z，y，x，x，z），则两者中最大的共同前缀为（x,y,y,z），在 A’=B’=空表，则 A=B；若 A’=空表，而 B’<>空表，或者两者均不为空表，且 A’的首元小于 B’的首元，则 A<B；否则 A>B。（词典次序）试写一个比较 A，B 大小的算法（在算法中，不要破坏原表 A 和 B，并且不一定先求得 A’和 B’才进行比较）。

36. 设有一个用向量表示的线性表 L，要求写出一个将该表逆置的过程，并允许在原表的存储空间外再增加一个附加的工作单元。(朱儒荣, C 语言版数据结构考研例题)

解：用数据类型描述 Seqlist 顺序存储结构：

```
vold converts(seqlist L)

{ k=L.length;

m = k/2;
```

```
for(i = 0;i<m;i++) {

x = L.element[i];

L.element[i] = L.element[k-i-1];

L.element[k-i-1] = x;

}

} // converts
```

讨论:这个算法过程只须进行数据元素的交换操作,其主要时间花费在for循环上,整个算法的时间复杂度为O(k)。