

浙江理工大学

二〇〇八年硕士学位研究生招生入学考试试题

考试科目：数据结构与数据库技术 代码：938

(* 请考生在答题纸上答题，在此试题纸上答题无效)

第一部分、数据结构(共 90 分)

一、选择填空题(每空格 3 分，本题共 60 分)

1. 已知单链表结点的存储结构如下：

```
struct node {
    int data;
    struct node *next;};
```

这里，单链表的头指针为 head，数据域为 data，指针域为 next。试在下列 A~J 中选择一个正确答案，填入相应的空格处，分别实现下列四小题的算法功能，注意各个小题之间没有联系。

1) 将单链表中值相同的多余结点删除。

```
void test1(struct node *head) {
    struct node *p,*q,*r;
    p=head;
    while (p!=NULL) {
        r=p;
        _____ (1) _____
        while (q!=NULL) {
            if (q->data==p->data) _____ (2) _____
            else r=q;
            q=q->next;
        }
        _____ (3) _____
    }
}
```

2) 将值为 y 的结点插入到值为 x 的结点之后，如果值为 x 的结点不存在，则将其插入到单链表的表尾。

```
void test2(struct node *head,int x,int y) {
    struct node *p,*q,*r;
```

```

r=(struct node *)malloc(sizeof(struct node));
r->data=y;
int sgn=0;
p=head;
while (p!=NULL)&&(sgn==0) {
    if (p->data==x) sgn=1;
    _____ (4) _____
    p=p->next;
}
if (sgn==1) _____ (5) _____
else _____ (6) _____
q->next=r;
}
}
    
```

3) 假设在上述单链表结点的存储结构中增加另一个指针域 prior, 将该单链表改造成双向循环链表(假设该单链表中至少有一个结点)。

```

void test3(struct node *head) {
    struct node *p,*q;
    p=head;
    while (p!=NULL) {
        q=p->next;
        if (q!=NULL) _____ (7) _____
        else {
            _____ (8) _____
            head->prior=p;
            break;
        }
        p=p->next;
    }
}
    
```

4) 若采用单链表结构去存储一个堆栈, 分别实现在堆栈中入栈和出栈一个元素的算法。

```

void test4(struct node *head,int x) {
    struct node *p;
    p=(struct node *)malloc(sizeof(struct node));
    p->data=x;
    
```

```
p->next=head;
    _____
    (9)
}
```

```
int test5(struct node *head) {
    struct node *p;
    if (head!=NULL) {
        p=head;
        _____
        (10)
        x=p->data;
        return(x); }
    else exit(1);
}
```

本题选项:

- (A) head=head->next; (B) head=p; (C) q=p; (D) q->prior=p;
 (E) q=p->next; (F) p->next=head; (G) p=p->next; (H) r->next=p;
 (I) r->next=q->next; (J) r->next=NULL;

2. 已知二叉树结点的链表存储结构如下:

```
struct node {
    char data;
    struct node *lch,*rch; };
```

这里，树结点的数据域为 data，左孩子指针域为 lch，右孩子指针域为 rch，根结点所在链结点的指针由 BT 给出。试在下列 A~J 中选择一个正确答案，填入相应的空格处，分别实现下列两个小题的算法功能，注意各个小题之间没有联系。

1) 利用中序遍历算法，查找二叉树 BT 中值为 x 的这个结点的双亲、左孩子及右孩子。

```
void test6(struct node *BT,char x)
{
    struct node *p,*q,*s[100];
    struct node *x1,*x2,*x3;
    int top=0;
    x1=x2=x3=NULL;
    p=BT;
    while ((top!=0)||(p!=NULL)) {
```

```

    if (p!=NULL) while(p!=NULL) {
        top++;
        s[top]=p;
        p=_____ (11)
    }
    else {
        p=s[top];
        _____ (12)
        if (p->data==x) {
            _____ (13) =p->lch;
            _____ (14) =p->rch;
        }
        q=p;
        p=p->rch;
        if (p!=NULL)&&(p->data==x) _____ (15) =q;
    }
}

if(x1!=NULL) printf("结点 x 的双亲是:%c\n",x1->data);
else printf("结点 x 是树根\n");
if(x2!=NULL) printf("结点 x 的左孩子是:%c\n",x2->data);
else printf("结点 x 无左孩子\n");
if(x3!=NULL) printf("结点 x 的右孩子是:%c\n",x3->data);
else printf("结点 x 无右孩子\n");
}

```

2) 假设上述二叉树 BT 为一个二叉排序树, 实现在该二叉排序树中插入一个结点 s 的算法。

```

void test7(struct node *BT,struct node *s)
{
    struct node *p,*q;
    if (BT==NULL) BT=s;
    else {
        _____ (16)
        while (p!=NULL) {
            q=p;
            if (s->data<p->data) _____ (17)
            else _____ (18)
        }
    }
}

```

```

    }
    if(s->data<q->data) _____ (19)
    else _____ (20)
    }
}
    
```

本题选项:

- (A) x1 (B) x2 (C) x3 (D) p=p->lch; (E) p=p->rch;
 (F) p->lch; (G) p=BT; (H) q->lch=s; (I) q->rch=s; (J) top--;

三、算法程序设计题（每小题 15 分，本题共 30 分）

1. 设哈希函数为 HT，解决冲突的方法为外链地址法，哈希函数采用除留余数法 ($k\%p$, k 为待删除的关键字, p 为小于基本哈希表容量 m 的质数)。若哈希表中某个位置上的 key 域值为零，则表示该位置未被占用。试编写程序，实现从用哈希表中删除关键字 k 的算法（注意需要判断关键字是否存在）。

```

define m 100;
struct node
{ int key;
  struct node *next;
}HT[m];
void test1(struct node HT[],int k,int p)
    
```

2. 试编写程序，实现用单链表表示的数据简单选择排序，并分析算法的时间复杂度。这里结点的数据域为 data，指针域为 next，单链表的头结点为 head。

```

struct node
{ int data;
  struct node *next; };
void test2(struct node *head)
    
```

第二部分、数据库技术(共 60 分)

解答题（本题共 8 小题，可以选择 6 小题解答，每小题 10 分，按得分最高的 6 小题计分。本题共 60 分）

数据库