

天津大学研究生院 2000 年招收硕士生入学试题

考试科目：数据结构与程序设计

题号：528

页数：8页

答题须知：

- 一、简答题要求：简明扼要，书写整齐。
- 二、填空题只填标号即可。
- 三、设计题要求：
 - (1) 给出数据结构的定义。
 - (2) 给出变量的类型说明和用途说明。
 - (3) 先使用程序流程图、N-S图、结构化汉语表达工具之一给出算法或过程设计，再用 C++ (含 C) 给出程序设计。
 - (4) 书写要整齐、清楚。

题一. 简答题

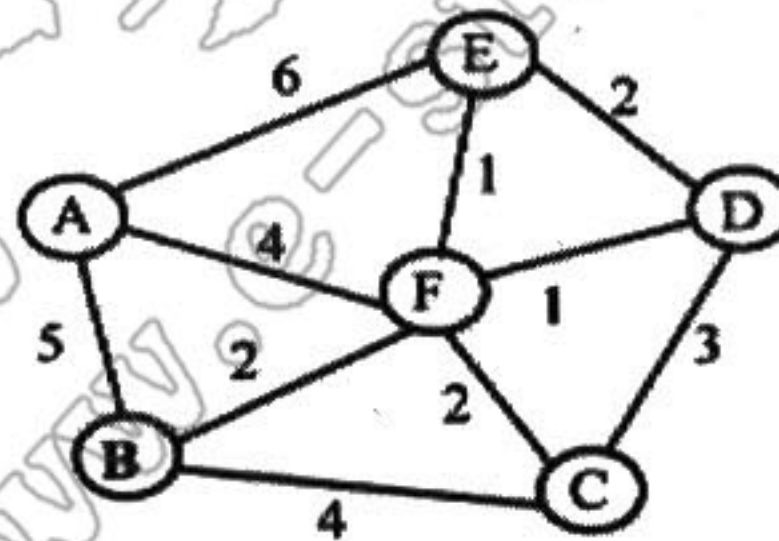
1.1 对 n 个顶点的无向图 G ，采用邻接矩阵表示，如何判别下列有关问题(6分)：

- (1) 图中有多少条边？
- (2) 任意两个顶点 i 和 j 是否有边相连？
- (3) 任意一个顶点的度是多少？

1.2 已知一棵二叉树的中序遍历和后序遍历的结果分别为 BDCEAFHG 和 DECBHGFA，画出这棵二叉树。(5分)

1.3 试写出递归的二分查找算法(不要求编程)。(5分)

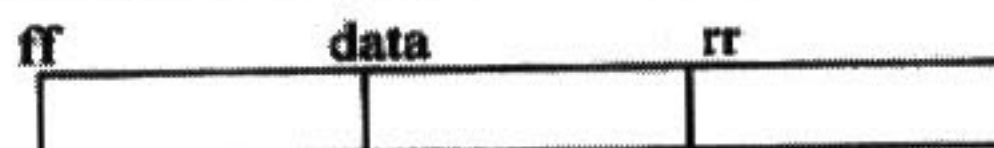
1.4 请以下图为例说明“最小生成树”的算法并画出该图的最小生成树。(7分)



题二. 设计题

2.1 请写出在已知双向链表中删除某指定结点的算法并用 C 编程。(7分)

假设结点的各域分别为 ff、data、rr，如下图所示：



2.2 试编写算法并用 C 编程, 判断两棵二叉树是否等价. (10 分)

称二叉树 T1 和 T2 是等价的: 如果 T1 和 T2 都是空的二叉树; 或者 T1 和 T2 的根结点的值相同, 并且 T1 的左子树与 T2 的左子树是等价的, T1 的右子树与 T2 的右子树是等价的。

题三. 选择填空(10 分):

3.1 使用操作符对数据进行格式输出, 应包含()文件:

- A. iostream.h B. iomanip.h C. stdio.h D. stdlib.h

3.2 下列函数中, ()不能重载:

- A. 成员函数 B. 非成员函数 C. 析构函数 D. 构造函数

3.3 下列运算符重载的描述中, ()是正确的:

- A. 可改变操作数个数 B. 可改变优先级 C. 可改变结合性
D. 不可改变语法结构

3.4 下列纯虚函数和抽象类的描述中, ()是错误的:

- A. 纯虚函数没有具体实现 B. 抽象类是具有纯虚函数的类
C. 抽象类的派生类一定不再是抽象类 D. 抽象类不能建立对象

3.5 下列动态联编的描述中, ()是错误的:

- A. 动态联编是在编译时确定操作函数的
B. 动态联编是以虚函数为基础的
C. 动态联编必须通过指针或引用来调用虚函数
D. 动态联编是在运行时确定操作函数的

3.6 派生类的构造函数的成员初始化列表中, 不能包含():

- A. 基类的构造函数 B. 派生类中子对象的初始化
C. 基类的子对象的初始化 D. 派生类中数据成员的初始化

3.7 转换构造函数应该是():

- A. 缺省构造函数 B. 不带参数的构造函数
C. 带一个参数的构造函数 D. 带两个以上参数的构造函数

3.8 下列各函数中, ()不是类的成员函数:

- A. 构造函数 B. 析构函数
C. 拷贝初始化构造函数 D. 友元函数

3.9 类 A 中的成员函数为 void Set(A&a), 其中 A&a 的含义是():

- A. a 是类 A 的对象引用, 用作函数 Set() 的形参
B. 变量 A 与 a 按位相与作为函数 Set() 的形参
C. 指向类 A 的指针为 a
D. 将 a 的地址赋给函数 Set

528

3.10 p 是指向类 A 数据成员 m 的指针, A1 是类 A 的一个对象,
() 是给 m 赋值 5:

- A. A1.p=5 B. A1.*p=5 C. A1->p=5 D. *A1.p=5

题四. 写出下面程序的运行结果(10 分):

4.1 (2.5 分):

```
#include <stdio.h>
int n = 1;
void func()
{
    static int a = 2;
    int b = 5;
    a += 2;
    n += 12;
    b += 5;
    printf("%d,%d,%d\n", a, b, n);
}
void main()
{
    static int a;
    int b = -10;
    b += 4;
    func();
    printf("%d,%d,%d\n", a, b, n);
    n += 10;
    func();
}
```

4.2 (2.5 分):

```
#include <stdio.h>
void main()
{
    static char t[3][5] = {"abcd", "efgh", "ijkl", "mnop", "qrst", "uvwxy"};
    printf("%c%c%c%c", t[1][2][3], *(t[1][2]+3), *((*(t+1)+2)+3), *((t[1]+2)+3));
    printf("%s,%s,%s,%s", **t, *((t+1)+1), t[0][2], **t+1);
}
```

4.3 (2.5 分):

```
#include <stdio.h>
void main()
{
    int val=5;
    int &refv = val;
```

057

2000125

```
    refv += 5;
    printf("%d,", val);
    int *p = &refv, val1 = refv;
    printf("%d,%d", *p, val1);
}
```

4.4 (2.5 分):

```
#include <stdio.h>
void fun(int a, int b, int *c)
{
    b += a;
    *c = b - a;
}

void main()
{
    int x, y, z;
    fun(5, 6, &x);
    fun(7, x, &y);
    fun(x, y, &z);
    printf("%d %d %d", x, y, z);
}
```

题五. 试针对本题中所给的程序，分析其中的链表操作并解答下列问题(10 分):

5.1 写出本题程序运行时，按序进行下面的运行输入之后的运行结果：

Data? 4
Data? 3
Data? 6
Data? 4
Data? -2

5.2 将函数 ShowList() 改成非递归函数

5.3 在主函数结束之前，增加一个删除整个链表的函数 DeleteList()，
从而释放内存空间。

附：本题设定的程序如下：

```
#include <stdio.h>
#include <alloc.h>
struct Lnode
{
    int data;
    struct Lnode *next;
};
```

528

```
void ShowList(struct Lnode *list)
{
    if(list)
    {
        printf("%d\n", list->data);
        if(list->next)
            ShowList(list->next);
    }
}

struct Lnode *GetNode()
{
    struct Lnode *item;
    item = malloc(sizeof(struct Lnode));
    if(item)
    {
        item->next = NULL;
        item->data = 0;
    }
    else printf("Nothing allocated!");
    return item;
}

void main()
{
    struct Lnode *head = NULL, *p1, *temp;
    int n = 0;
    p1 = temp = GetNode();
    while(temp != NULL)
    {
        printf("data? ");
        scanf("%d", &temp->data);
        if(temp->data > 0)
        {
            n++;
            if(n == 1) head = temp;
            else p1->next = temp;
            p1 = temp;
        }
        else
            break;
        temp = GetNode();
    }
    ShowList(head);
}
```


题六. 下面两小题任选一题对程序进行分析并回答问题 (若两题全选只第一题有效)：

6.1 (5 分):

```
#include <iostream.h>
class A
{
    public:
    virtual void act1();
    void act2() { act1();}
};

void A::act1()
{
    cout << "A::act1() called." << endl;
}

class B: public A
{
    public:
    void act1();
};

void B::act1()
{
    cout << "B::act1() called." << endl;
}

void main()
{
    B b;
    b.act2();
}
```

回答下列问题:

- (A) 该程序执行后的输出结果是什么? 为什么?
(B) 如果将 A::act2() 的实现改为如下, 输出结果是什么?

```
void A::act2()
{
    this->act1();
}
```

- (C) 如果将 A::act2() 的实现改为如下, 输出结果是什么?

```
void A::act2()
{
    A::act1();
}
```

528

01

6.2 (5 分):

```
#include <iostream.h>
```

```
class A
```

```
{
```

```
    public:
```

```
        void f1();
```

```
    protected:
```

```
        int j1;
```

```
    private:
```

```
        int i1;
```

```
};
```

```
class B: public A
```

```
{
```

```
    public:
```

```
        void f1();
```

```
    protected:
```

```
        int j2;
```

```
    private:
```

```
        int i2;
```

```
};
```

```
class C: public B
```

```
{
```

```
    public:
```

```
        void f3();
```

```
};
```

回答下列问题:

- (A) 派生类 B 中成员函数 f2() 能否访问基类 A 中成员 f1(), i1, j1?
- (B) 派生类 B 的对象 b1 能否访问基类 A 中成员 f1(), i1, j1?
- (C) 派生类 C 中成员函数 f3() 能否访问直接基类 B 中成员 f2(), i2, j2?
派生类 C 中成员函数 f3() 能否访问间接基类 B 中成员 f1(), i1, j1?
- (D) 派生类 C 的对象 c1 能否访问直接基类 B 中成员 f2(), i2, j2?
派生类 C 的对象 c1 能否访问间接基类 A 中成员 f1(), i1, j1?
- (E) 总结公有继承的特性

题七. Josephus 问题 (15 分):

NumOfBoys 个小孩围成一圈，从第一个小孩起，顺时针方向数数，每数到第 interval 个小孩时，该小孩便离开。小孩不断离开，圈子不断缩小，直至剩下最后一个小孩。试分别使用数组结构和环链结构，设计相应算法后编写程序以便找到剩下的最后一个小孩。

061

2000129

题八. 编写一个类 `stack`，利用链表结构实现简单的栈(10 分):

其中:

成员函数为:

```
void stack::push(int item); // 将数据 item 插入到栈中
int  stack::pop();          // 从栈中取一个数据
```

数据成员为:

一个指向链首的指针

链表结构为:

```
struct Node
{
    int a;
    struct Node *next;
};
```

使用对象的过程为:

```
stack stc;
```

```
stc.push(1);
```

```
stc.push(2);
```

```
stc.push(3);
```

```
cout << stc.pop() << endl; // 输出 3, 栈中余 1,2
```

```
cout << stc.pop() << endl; // 输出 2, 栈中余 1
```