

# 华侨大学 2013 年硕士研究生入学考试专业课试卷

(答案必须写在答题纸上)

招生专业 计算机技术

科目名称 数据结构与 C++ 科目代码 850

## 第一部分 数据结构 (共 75 分)

### 一、单项选择题 (每小题 2 分, 共 24 分)

1. 执行下面程序段时, 则 S 语句的语句频度是 ( )。

```
for (int i =1;i<=n;i++)  
    for (int j =1; j <=i ; j++)  
        S;
```

A)  $n^2$             B)  $n^2/2$             C)  $n(n+1)$             D)  $n(n+1)/2$

2. 在一个不带头结点的单链表 HL 中, 若要向表头插入一个由指针 p 指向的结点, 则执行 ( )。

A) HL=p; p->next=HL;            B) p->next=HL; HL=p;  
C) p->next=HL; p=HL;            D) p->next=HL->next; HL->next=p;

3. 设栈 S 和队列 Q 的初始状态皆为空, 元素 a1, a2, a3, a4, a5 和 a6 依次通过一个栈, 一个元素出栈后即进入队列 Q, 若 6 个元素出队列的顺序是 a3, a5, a4, a6, a2, a1 则栈 S 至少应该容纳 ( ) 个元素。

A) 3            B) 4            C) 5            D) 6

4. 一棵树高为 K ( $K \geq 1$ ) 的完全二叉树至少有 ( ) 个结点。

A)  $2^{k-1}$             B)  $2^k$             C)  $2^{k+1}$             D)  $\log_2 k$

5. 由权值分别为 3, 8, 6, 2, 5 的叶子结点生成一棵赫夫曼树, 它的带权路径长度为 ( )。

A) 24            B) 48            C) 72            D) 53

6. 假设一个有 n 个顶点和 e 条弧的有向图用邻接表表示, 则删除与某个顶点  $v_i$  相关的边的时间复杂度是 ( )。

A)  $O(n)$             B)  $O(e)$             C)  $O(n+e)$             D)  $O(n*e)$

7. 图的深度优先遍历类似于树的 ( )。

A) 先序遍历            B) 中序遍历            C) 后序遍历            D) 层次遍历

8. 对一棵二叉排序树进行 ( ) 遍历, 可以得到该二叉树的所有结点按值从小到大排列的序列。

A) 前序            B) 后序            C) 中序            D) 按层次

9. 设散列表长  $m=14$ , 散列函数  $H(K) = K \% 11$ , 已知表中已有 4 个结点:  $r(15)=4$ ;  $r(38)=5$ ;  $r(61)=6$ ;  $r(84)=7$ , 其他地址为空, 如用二次探测再散列处理冲突, 关键字为 49 的结点地址是 ( )。

A) 8            B) 3            C) 5            D) 9

10. 用某种排序方法对关键字序列 (23, 72, 21, 47, 15, 27, 59, 35, 20) 进行排序时,

前两趟的结果情况如下：

15, 72, 21, 47, 23, 27, 59, 35, 20

15, 20, 21, 47, 23, 27, 59, 35, 72

则所采用的排序方法是 ( )。

A) 选择排序    B) 起泡排序    C) 归并排序    D) 快速排序

11.  $n$  个顶点的无向图中含有边的数目最多为 ( )。

A)  $n-1$     B)  $n$     C)  $n(n-1)/2$     D)  $n(n-1)$

12. 由权值分别为 11, 8, 6, 2, 5 的叶子结点生成一棵赫夫曼树, 它的带权路径长度 WPL 为 ( )。

A) 24    B) 71    C) 48    D) 53

## 二、简答与应用题 (30 分)

1. (8 分) 假设链队列带头结点, 其出队操作如下:

```
Status DnQueue_L(LinkQueue &Q ,ElemType &e) {  
    //链队列 Q 出队, 出队元素赋值给 e 中  
    if (Q.front == Q.rear) return ERROR;  
    p = Q.front->next; e = p->data;  
    Q.front->next = p->next;  
    if(Q.rear == p) Q.rear = Q.front; //问题 (1)  
    free(p);  
    return OK;  
} // DeQueue_L
```

(1) (4 分) 此处的 if 语句起什么作用 (含义)?

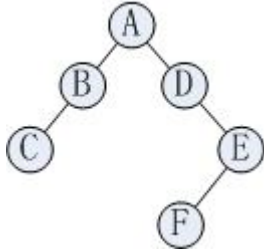
对出队前若队列里只有一个结点的情况处理, 此时需要修改 Q.rear 值。

(2) (4 分) 假设链队列不带头结点, 请补充下面的代码, 以实现类似的链队列出队操作。

```
Status DnQueue_L(LinkQueue &Q ,ElemType &e) {  
    //不带头结点的链队列 Q 出队, 出队元素赋值给 e 中  
    if (Q.front==NULL) return ERROR; //空队列  
    if (Q.front==Q.rear) { //只有一个结点的处理  
        e=Q.front.data;  
        p=Q.front;  
        _____(1)_____;  
        free(p); return OK;  
    }  
    p = Q.front; e = p->data;  
    _____(2)_____;  
    ;  
    free(p); return OK;
```

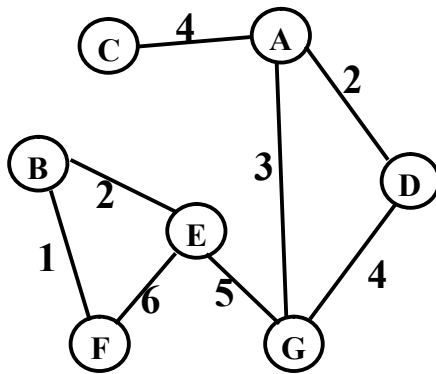
```
}// DeQueue_L
```

2. (8分) 请对下图所示的二叉树进行中序线索化, 为每个空指针建立相应的前驱和后继, 请用虚线箭头在图中画出结果。



3. (8分) 考虑无向网G:

- (1) (2分) 给出邻接表存储结构(要求邻接表的每个顶点的邻接链表按结点域升序排列, 每一表结点包含所表示的边的权值)。
- (2) (2分) 请给出从顶点A开始的深度优先顶点访问序列(要求根据邻接表进行遍历)。
- (3) (4分) 根据克鲁斯卡尔(Kruskal)算法, 画出无向图G的最小生成树。



无向网 G

4. (6分) 有一组关键字序列 {26, 38, 73, 21, 54, 35, 167, 32, 7, 223, 52}, 试用哈希函数  $H(\text{key}) = \text{key} \bmod 13$  和链地址(拉链)法解决冲突方法构造这组关键字的哈希表(设哈希表的长度为15)。

### 三、算法设计题 (21分)

1. (10分) 设单向链表的类型定义如下:

```
typedef int ElemType; //设表中元素类型为 int
typedef struct LNode{
    ElemType data;
    Node* next;
}*LinkedList;
```

编写函数 `LinkedList CopyListFrom(LinkedList list)`，返回已知的带头结点单链表 `list` 的一个拷贝（备份）链表。

2. (11 分) 设二叉排序树 `bt` 以二叉链表为存储结构，试设计算法删除二叉排序树 `bt` 中值最大的结点。

## 第二部分 C++ (共 75 分)

### 四、单项选择题 (每小题 2 分, 共 20 分)

1. 对于结构体中定义的成员，其默认访问权限为 ( )。

- A) `public`      B) `protected`      C) `private`      D) `static`

2. 若有定义：`int m=10;`，下面的引用声明中，( ) 是正确的。

- A) `int &x=m;`      B) `int& y=10;`      C) `int&z=int;`      D) `int& w=NULL;`

3. 若有定义：`enum Color{ blue=1, yellow, green, white=8, black, red } favorite;`，以下正确的语句是 ( )。

- A) `blue++;`      B) `int k=black+1;`      C) `favorite=7;`      D) `favorite++;`

4. 若已有定义：`struct Student{ char *name, int age, float mark;}s[8]; Student* p=s+3;`，以下语句(组)，正确的是 ( )。

- A) `cin>>p[0].name>>p[0].age>>p[0].mark;`  
B) `cin>>p->age>>p->mark; p->name="Liumei";`  
C) `p->name=new char[10]; p->name="Liumei";`  
D) `s->name=new char[10]; strcpy(p->name, "Liumei");`

5. 下面程序运行时，输出“`data=10`”，下划线空白处应填入的正确语句是 ( )。

```
#include<iostream>
using namespace std;
class Myclass{
    int data;
public:
    Myclass(int value){ data=value;}
    void SetNum(int newvalue){ data=newvalue; }
    void print(void){ cout<<"data="<<data<<endl; }
};
void main(void){
    Myclass *p;
    p->print();
}
```

- A) `p=Myclass(10);`      B) `p->SetNum(10)`      C) `p.Myclass(10)`      D) `p=new Myclass(10);`

6. 下面叙述错误的是 ( )。

- A) 类的 private 成员不能被该类的对象直接访问
- B) 派生类的对象不可能直接访问基类的 public 的成员
- C) 类的友元函数可以访问类中的私有成员
- D) 指向基类的指针可以指向派生类的对象

7. 派生类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) 派生类中成员对象的初始化
- C) 基类中成员对象的初始化
- D) 派生类中一般数据成员的初始化

8. 若有以下源程序代码, 则 ( )。

```
#include<iostream>
using namespace std;
class Myclass{
    int data;
public:
    Myclass() { cout<<"constructor is called."<<endl; data=0; }
    ~Myclass() { cout<<"desstructor is called."<<endl; }
};
void main(void) {
    Myclass obj[3];
}
```

- A) 编译出错
  - B) 输出 1 行 "constructor is called." 和 1 行 "desstructor is called."
  - C) 编译没有错误, 运行时出错
  - D) 输出 3 行 "constructor is called." 和 3 行 "desstructor is called."
9. 若对上面的类 Myclass 添加一个拷贝(复制)构造函数(copy constructor), 则该拷贝构造函数的定义, 不正确的是 ( )。

- A) 

```
Myclass(Myclass x) {
    data=x.data;
    cout<<"copy constructor is called."<<endl;
}
```
- B) 

```
Myclass(Myclass &x) {
    data=x.data;
    cout<<"copy constructor is called."<<endl;
}
```
- C) 

```
Myclass(const Myclass *x) {
    data=x->data;
    cout<<"copy constructor is called."<<endl;
}
```

```
D) Myclass(const Myclass& x) {  
    data=x.data;  
    cout<<"copy constructor is called."<<endl;  
}
```

10. 类的友元函数的作用是 ( )。

- A) 增加成员函数的种类  
B) 增强类的封装性  
C) 提高程序的运行效率  
D) 建立类之间的继承关系

### 五、阅读以下程序，回答相关问题（28分）

1. (6分) 写出下面程序的运行结果。

```
#include<iostream>  
using namespace std;  
void Fun(int *p1, int *p2) {  
    (*p1)++;  p2++;  
    cout<<"In Fun:*p1="<<*p1<<endl;  
    cout<<"In Fun:*p2="<<*p2<<endl;  
}  
void main(void) {  
    int data[5]={1, 2, 3, 4, 5}, *p=data, *q=data+1;  
    Fun(p, q);  
    cout<<"In main:*p="<<*p<<endl;  
    cout<<"In main:*q="<<*q<<endl;  
}
```

2. (6分) 写出下面程序的运行结果。

```
#include<iostream>  
using namespace std;  
class CShape {  
public:  
    virtual float area() { return 0.0; }  
};  
class CTriangle:public CShape {  
public:  
    CTriangle(float h=0, float w=0) { H=h; W=w; }  
    float area() { return 0.5*H*W; }  
private:  
    float H, W;  
};  
class CCircle:public CShape {  
public:
```

```

    CCircle(float r=0) { R=r; }
    float area() { return 3.14*R*R; }
private:
    float R;
};
void main(void) {
    CShape shape;
    cout<<"shape.area()="<<shape.area()<<endl;
    CTriangle tri(3,4);
    cout<<"tri.area()="<<tri.area()<<endl;
    CCircle cir(5);
    cout<<"cir.area()="<<cir.area()<<endl;
    CShape* s1=&tri;
    cout<<"s1->area()="<<s1->area()<<endl;
    CShape& s2=cir;
    cout<<"s2.area()="<<s2.area()<<endl;
}

```

3. (8分)

```

#include<iostream>
using namespace std;
class Student{
    char* name;
    int age;
    float score;
public:
    Student(char* n,int a,float s):age(a),score(s){
        cout<<"constructing..."<<n<<endl;
        name=new char[strlen(n)+1];
        strcpy(name,n);
    }
    Student(const Student& s){
        cout<<"copy nstructing..."<<s.name<<endl;
        name=new char[strlen(s.name)+1];
        strcpy(name,s.name);
        age=s.age;
        score=s.score;
    }
    ~Student(){

```

```

        cout<<"destructing..."<<name<<endl;;
        delete[] name;
    }
};
void main(void) {
    Student s1("Liu", 18, 86.5), s2=s1;
}

```

(1) (4分) 给出程序的运行结果;

(2) (4分) 若把成员函数 Student(const Student& s)的定义去掉, 程序还能正常运行吗? 给出简要理由。

4. (8分)

```

#include<iostream>
using namespace std;
class Base{
public:
    int Bvalue;
    Base(int v){
        Bvalue=v;
    }
};
class D1: public Base{
public:
    int D1value;
    D1(int v, int b):Base(v){
        D1value=b;
    }
};
class D2: public Base{
public:
    int D2value;
    D2(int v, int c):Base(v){
        D2value=c;
    }
};
class D: public D1, public D2{
public:
    int Dvalue;
    D(int a, int b, int c, int d):D1(a, b), D2(c, d){

```



```

        Dvalue=d;
    }
};

void main(void) {
    D obj(1, 2, 3, 4);
    cout<<obj.Bvalue<<endl; //问题 (1)
}

```

(1) (4 分) 程序能编译通过吗? 给出简要理由;

(2) (4 分) 若将类 Base 说明为虚基类, 上面程序需要做哪些相应的修改?

## 六、编程题 (27 分)

1. (15 分) 请补充完成带头结点的单链表类 LinkList 中构造函数和析构函数的定义。

```

#include<iostream>
using namespace std;
struct Node{ //单链表中的结点类型
    int data; //假设表中元素类型为 int
    Node* next;
};
class LinkList{ //带头结点的单链表类的定义
    Node* head;
public:
    LinkList(int data[],int n){// (1) 补充完成该构造函数的定义 (8 分)
        //建立一个由 n 个元素 (data[0]~data[n-1]) 构成的带头结点的单链表
        ?
    }//LinkList
    LinkList(void) { //建立一个空的带头结点的单链表
        cout<<"constructing an empty linked list."<<endl;
        head=new Node;
        head->next=NULL;
    }//LinkList
    void Print(void) { //输出链表中的各元素值
        cout<<"The content of the list:";
        if(head->next==NULL) {
            cout<<"Empty linked list!"<<endl;
            return;
        }
        Node* t=head->next;
        while(t!=NULL) { cout<<t->data<<" "; t=t->next; }
    }
};

```

```

        cout<<endl;
    }//Print

    ~LinkedList() { // (2) 完成该析构函数的定义 (7分)
        ?
    }//~LinkedList
}; //类LinkedList 定义结束
void main(void) {
    int a[]={1,2,3,4,5};
    int size=sizeof(a)/sizeof(int);
    LinkedList mylist(a, size);
    mylist.Print();
    cout<<endl;
}

```

2. (12分) 编写程序，输出如下杨辉三角数字图形。

```

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
1 9 36 84 126 126 84 36 9 1

```