

## 天津大学研究生院 2003 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: 465  
页数: 共 5 页

### 一. 数据结构 (75分)

1. 试简述广义表与线性表的定义。(6分)

2. 有算术表达式  $A-B+C/D+E \#$ , 其中  $\#$  号为表达式结束符, 试画出对其求值的操作数栈和运算符栈的逐步变化过程。(7分)

3 将算术表达式  $((A+B)+C*(D+E)+F)*(G+H)$  转化为二叉树表示, 并给出其逆波兰(即后序遍历)表示。(8分)

4. 假定用于通信的电文仅由 8 个字母  $c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8$  组成, 各字母在电文中出现的频率分别为: 5, 25, 3, 6, 10, 11, 36, 4. 试用这 8 个字母构造哈夫曼树, 并设计 Huffman 编码, 再给出该电文字母的总码数。(10分)

5. 对一组关键字, 按哈希函数  $H(\text{Key}) = \text{Key} \text{ MOD } 13$  和线性探测处理冲突技术构造所得哈希表  $a.\text{elem}[0..15]$ , 表长  $m=16$

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	14	01	68	27	55	19	20	84	79	23	11	10			

分别给出  $K = 84, K = 38$  的查找过程(对查找结果应说明成功或失败)。(10分)

6. 试修改起泡排序算法, 在正反两个方向交替进行扫描, 即第一趟把排序码最大的对象放到序列的最后, 第二趟把排序码最小的对象放到序列的最前面。如此反复进行。(8分)

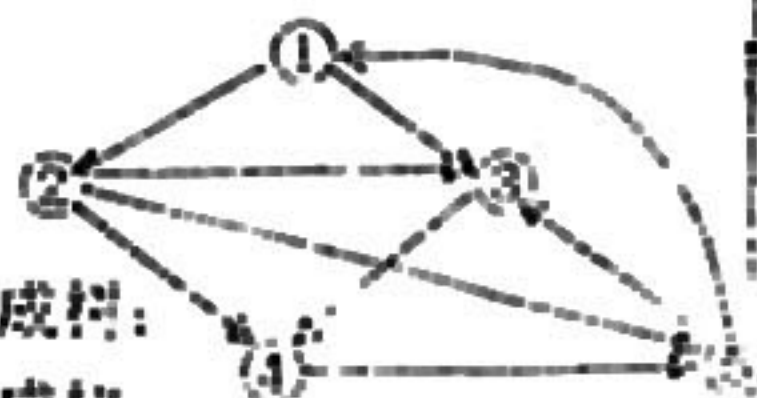
7. 对于如右图所示的有向图

(1) 给出右图的邻接表存储表示:

(2) 从顶点①出发进行深度优先搜索所得到的深度优先生成树:

(3) 从顶点②出发进行广度优先搜索所得到的广度优先生成树:

(4) 设该有向图存储在邻接表中, 试设计一个算法, 按深度优先搜索策略对其进行拓扑排序。(16分)





8. 试编写递归算法判断两棵二叉树是否等价。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 的左子树和 T2 的左子树是等价的，T1 的右子树和 T2 的右子树是等价的。(10分)

一 程序填空 (每题 3 分，共 20 分)

1 为下面的程序片段填空，使得(1),(2)的结果相同。

(1) if \_\_\_\_\_ min=a;  
 else if \_\_\_\_\_ min=b;  
 else min=c;

(2) min=a;  
 if \_\_\_\_\_ min=b;  
 if (min>c) min=c;

2 下面程序在屏幕上输出小于 200 的素数，请完成程序填空。

```
class Prime {
    int p;
    public:
    Prime(int n) {if (n<3) p=3; else p=n;};
    int Run { int k,j,flag;
    for(j=2;j<p;j++)
    { _____;
    for(k=2; k<j; k++)
    if (_____) flag=0;
    if (_____) cout<<j<<endl; } };
    int main() { Prime obj(200); _____; }
```

3 下面程序将链首删除，完成该程序

```
class node { public: int dt; node *lk; node(int n) {dt=n;lk=NULL;};
class list {node *head;
    public: list(int a[],int n){int j, *p; if(n<1) {head=0;return;}
        head=new node(a[0]);
        p=head;
        for(j=1;j<n;j++){p->lk=new node(a[j]); _____; }
        int del() { node *p=head; if(_____) {head=head->lk;delete p;}}
        ~list(){node *p=head,*q; while(_____) {q=p; _____;delete p;}}.
    int main()
    {int a[12]={1,2,3,4,5,6,7,8,9,10,11,12};
    list alist(a,12); alist.del();}
```

4. Armstrong 数是指一个 n 位数其每位数的 n 次幂的和，例如 153 是 3 位的 Armstrong 数  
 完成下面的程序求出小于 999 的其他 3 个 Armstrong 数。

465

```
#include <iostream.h>
int main()
{
    const int n=3,base=10;
    unsigned int start_num, end_num, num, sum, power;
    int i, j, bit;
    unsigned int bit_power;
    power=1;
    for(i=1; i<=n; i++) power=power*base;
    start_num=_____ ; end_num=_____ ;
    for(num=start_num; num<=end_num; num++) {
        sum=0; power=10;
        for(i=1; i<=n; i++) {bit=_____ ;
            bit_power=1;
            for(j=1; j<=n; j++) bit_power=bit_power*bit;
            sum=_____ ; power=_____ ; }
        if (sum==num) {cout<<"\nthe num<<num<<"is an Armstrong num!; }
    } return 1; }
```

二、读下面的程序，给出结果(每题8分，共32分)

```
1
#include<iostream.h>
class M
{
    public: M(){a=b=0;}
           M(int i,int j){a=i; b=j;}
           void print(){cout<<a<<","<<b<<endl;}
    private: int a,b; };
void main()
{
    M m[2][4]; int x=10,y=10;
    for(int i=0;i<2;i++) for(int j=0;j<4;j++) m[i][j]=M(i+=2,j+=10);
    M(*pm)[4](m);
    for(int i=0;i<2;i++) { cout<<endl;
        for(int j=0;j<4;j++) (*(pm+i+j)).print(); }
    cout<<endl; }
```



2

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

```
class B1
```

```
{ public: B1(int i) { b1=i; cout<<"constructor B1."<<i<<endl; }  
    void print() { cout<<b1<<endl; } private: int b1;};
```

```
class B2
```

```
{ public: B2(int i) { b2=i; cout<<"constructor B2."<<i<<endl; }  
    void print() { cout<<b2<<endl; } private: int b2;};
```

```
class B3
```

```
{ public: B3(int i) { b3=i; cout<<"constructor B3."<<i<<endl; }  
    int getb3() { return b3; } private: int b3;};
```

```
class A:public B2,public B1
```

```
{ public: A(int i,int j,int k,int l):B1(i),B2(j),bb(k)  
    { a=i; cout<<"constructor A."<<i<<endl; }  
    void print() { B1::print(); B2::print(); cout<<a<<" "<<bb.getb3()<<endl; }  
    private: int a; B3 bb;};
```

```
void main()
```

```
{ A aa(1,2,3,4); aa.print(); }
```

3

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

```
static char i=0x42;
```

```
class A
```

```
{ public: virtual ~A() { i++; cout<<"A::~~A("<<i<<" ) called.\n"; } };
```

```
class B:public A
```

```
{ public: B(int i){ buf=new char[i]; }  
    virtual ~B() { char j=32; j+=i; delete [ ] buf;  
    cout<<"B::~~B("<<--j<<" ) called.\n"; }  
    private: char *buf;};
```

```
void fun(A *a)
```

```
{ delete a; }
```

```
void main()
```

```
{ A *a=new B(15);  
    fun( a); }
```

题号：465

4

```
#include <iostream.h>
#include <string.h>
class Base
{ public: void set_i(int x){i=x;}
  int get_i() {return i;}
protected: int i;
};
class Derived: private Base
{ public: Base::set_i;
  void set_j(int x) {j=x;}
  int get_ij() {return (j--)*(++i);}
protected: int j; };
int main()
{ Derived obj;
  obj.set_i(5);
  obj.set_j(7);
  cout<<obj.get_ij()<<endl;
  return 1;
}
```

三、程序设计题 (共 15 分)

设计一可以动态分配的串类，我们希望这种串类型是动态分配，能表示任意长度的串，该串类至少包含赋值，连接，复制等操作，其中，连接是将 2 个已知串连接起来产生新的串，复制也是对任意两串而言。最后编一主函数检查它们。在程序中需要使用友元函数。提示：一个串由指向字符的指针及串长度来描述。