

# 南开大学 2011 年硕士研究生入学考试试题

学 院：034 信息技术科学学院

考试科目：822 程序设计与数据结构

专 业：计算机技术

注意：请将所有答案写在专用答题纸上，答在此试题上无效！

## 程序设计部分

### 一、选择题（共 20 分）

1. （2 分）`char c='c'; char *p=&c; cout<<&p;` 输出的结果是（ ）
  - A. 'c'
  - B. &c
  - C. c
  - D. 以上都不是
2. （2 分）设有代码“`int x=5;`”，执行表达式语句“`!(++x)&&(x*=x);`”后，x 的值为（ ）
  - A. 5
  - B. 6
  - C. 25
  - D. 36
3. （2 分）下面的 C++关键字中，哪个不能用于变量的说明语句？（ ）
  - A. `extern`
  - B. `int`
  - C. `static`
  - D. `inline`
4. （2 分）关于虚函数的描述中，（ ）是正确的
  - A. 虚函数是一个 `static` 类型的成员函数
  - B. 虚函数是一个非成员函数
  - C. 基类中说明了虚函数后，派生类中与其对应的函数可不说明为虚函数
  - D. 派生类的虚函数与基类的虚函数具有不同的参数个数和类型
5. （2 分）下面关于类继承的描述中，哪个是正确的（ ）
  - A. 私有继承是将基类的成员继承为派生类的私有成员
  - B. 虚基类是只用来进行派生的类
  - C. 基类中的友元函数不能被派生类继承，友元类可以被继承
  - D. 函数重载时，除了名称重用之外，参数表和返回值类型不允许同时重用
6. （2 分）下列引用的定义，正确的是（ ）
  - A. `int &a = new int;`
  - B. `int &a = new int; a = 10;`
  - C. `int &a = *new int(10);`
  - D. `int & *a = new int;`

7. (2分) 定义动态一维数组: `int *p = new int[10];` 下列释放动态内存的语句正确的是 ( )
- A. `delete *p;`
  - B. `delete p[];`
  - C. `delete []p;`
  - D. `delete p[10];`
8. (2分) 关于类和对象的描述, 下列说法正确的是 ( )
- A. 类的友元函数可以访问该类的私有数据成员
  - B. 不同对象的数据成员共享相同的内存空间
  - C. 类的公有函数成员不能访问类的私有数据成员
  - D. 可以在定义类时为该类的数据成员赋初值
9. (2分) 对于变量说明 “`int a1 = 10, a2 = 3; float a3 = 8.0;`”; 表达式 `a1/a2*a3+a2/a1*a3` 的值为 ( )
- A. 0.454167
  - B. 26.6667
  - C. 29.0667
  - D. 24
10. (2分) 以下叙述不正确的是 ( )
- A. 全局变量在任何函数体内都有效
  - B. 函数的形式参数是局部变量
  - C. 静态变量的生存期为整个程序
  - D. 在函数体内的某个复合语句中定义的变量在本函数范围内有效

二、写出下列程序在 VC 6.0 环境下(X86 平台)的运行结果 (共 25 分)

1. (5分)

```
#include<iostream.h>
#include<string.h>
char* TestFunc(char* str){
    int i=0,j=0;
    char *fstr = new char[strlen(str)+1];
    while(str[i]!='\0') {
        if(str[i]>'0'&&str[i]<'9') {
            fstr[j]=str[i];
            j++;
        }
        i++;
    }
```

```

    }
    fstr[j]='\0';
    return fstr;
}
void main()
{
    cout<<TestFunc("#<a0b2cdef468>")<<endl;
    cout<<TestFunc("(1g3h57jk9)*")<<endl;
}

```

在 VC6.0 环境下执行结果:

## 2. (5 分)

```

#include<iostream.h>
void main(){
    int a[5][5] = {0},*p[5],i,j;
    for(i=0;i<5;i++){
        p[i]=a[i];
        for(i=0;i<5;i++){
            *(p[i]+i)=1;
            *(p[i]+4-i)=1;
        }
    }
    for(i=0;i<5;i++){
        for(j=0;j<5;j++)
            cout<<"t"<<p[i][j];
        cout<<endl;
    }
}

```

在 VC6.0 环境下执行结果:

## 3. (5 分)

```

#include<iostream.h>
void main()
{
    int i:=5;
    for(int i=1;i<=n;i++)

```

```

{
    int result = 1;
    for(int j=1;j<=i;j++);
    {
        result *=j;
    }
    cout<<"result = "<<result<<endl;
}
}

```

在 VC6.0 环境下执行结果:

4. (5 分)

```

#include <iostream.h>
void f(int x, int &y, const int &z)
{
    x+=z;
    y+=z;
    cout<<"x="<<x<<","y="<<y<<","z="<<z<<endl;
}
void main()
{
    int a=22,b=33,c=44;
    f(a,b,c);
    cout<<"a="<<a<<","b="<<b<<","c="<<c<<endl;
    f(a+b,b-c,2c-10)
}

```

在 VC6.0 环境下执行结果:

5. (5 分)

```

#include<iostream.h>
class TestClass
{
    static int count;
public:
    TestClass(){
        count = count +1;
    }
}

```

```

        cout<<"Construction, and count="<<count<<endl;
        if(count>1)
            cout<<"Warning: more than one instance!"<<endl;
    }
    ~TestClass(){
        count = count - 1;
        cout<<"count="<<count<<endl;
    }
};

int TestClass::count = 0;

void main()
{
    cout<<"Program begin..."<<endl;
    TestClass obj1;
    TestClass obj2;
    cout<<"Program end..."<<endl;
}

```

在 VC6.0 环境下执行结果:

### 三、根据题意要求,完成下列程序 (共 30 分)

1. (6 分) 函数 Square ()测试输入的参数是否是完全平方数, 如是返回 1 否则返回 0 (例如对 1,4, 9, 16, 25 返回 1)。请完善该函数。

```

int Square(int n)
{
    ____ (1) ____;
    while ( ____ (2) ____ )
        ++i;
    if ( ____ (3) ____ ) return 1;
    else return 0;
}
(1)
(2)
(3)

```

2. (6 分) 函数 Count ()计算字符串 str1 在字符串 str 中出现的次数。请完善该函数。

```

int Count(char str[], char str1[])
{
    int i,j,k,num=0;

```

```

for ( i=0; ____ (1) ____; i++)
    for ( ____ (2) ____, k=0; str1[k]==str[j]; k++; j++)
        if (str1 [ ____ (3) ____ ]==' \0' )
        {
            num++;
            i+=k;
            break;
        }
return num;
}
(1)
(2)
(3)

```

3. (6分) 定义一个函数模板 Sort, 实现对整数、浮点数和字符由小到大排序的功能。在主函数中定义了整型数组, 调用 Sort 函数完成对该数组的排序。请完善该程序。

```

#include<iostream.h>
_____(1)_____
void Sort(T *array, int size) {
    for(int i=0; i<size-1; i++)
        for(int j=size-1; j>i; j--)
            if( _____(2)_____ ) {
                int temp = *array;
                *array = *(array-1);
                *(array-1) = temp;
            }
}
void main()
{
    int a[10]={10,9,8,7,6,5,4,3,2,1};
    _____(3)_____
    for(int i=0; i<10; i++)
        cout<<a[i]<<" ";
    cout<<endl;
}
(1)
(2)
(3)

```

4. (6分) 设计程序, 计算 1000 以内的素数并保存到文本文件 prime.txt 中, 文件中的素

数以表格的形式输出（每行输出 5 个素数，每一个素数占用 5 个字符的宽度）。请完善该程序。

```
#include<fstream.h>
#include<iomanip.h>
void main()
{
    _____(1)_____;
    if(!fout){
        cout<<"File open error!"<<endl;
        return;
    }

    int n=1;
    for(int i=2;i<=1000;i++){
        for(int j=2,k=i/2;j<=k;j++){
            if(i%j==0)
                break;
            if(j>k){
                _____(2)_____;
                if(n++==5){
                    fout<<endl;
                    n=1;
                }
            }
        }
        _____(3)_____;
    }
    (1)
    (2)
    (3)
```

5. （6 分）建立一个由 lin 行、col 列构成的动态二维数组，并将该数组元素按行列输出。请完善该程序。

```
#include <iostream.h>
void main() {
    int lin,col,i,j;
    cout<<"lin,col=";
    cin>>lin>>col;
    int **b;
    _____(1)_____
    for(i=0;i<lin;i++)
```



```

        _____(2)_____
    for(i=0;i<lin;i++)
        for(j=0;j<col;j++)
            b[i][j]=i+j;
    for(i=0;i<lin;i++) {
        for(j=0;j<col;j++)
            _____(3)_____
        cout<<endl;
    }
}
(1)
(2)
(3)

```

### 数据结构部分

#### 四、选择题（共 24 分）

- （6 分）用数组保存 FIFO 队列：当删除元素时，队首指针向后移动；当有新元素加入时，若数组末端有空位，则将元素直接放置于队尾元素之后，否则移动元素使队首位于数组头，再加入新元素。则删除操作最好情况时间复杂性为\_\_\_\_\_，最坏情况时间复杂性为\_\_\_\_\_；添加操作最好情况时间复杂性为\_\_\_\_\_，最坏情况时间复杂性为\_\_\_\_\_， $n$  为队列元素总数。
  - $O(1)$
  - $O(\log n)$
  - $O(n)$
  - $O(n \log n)$
- （2 分）Shell 排序性能优于插入排序的原因是\_\_\_\_\_。
  - 通过间隔插入排序预处理，使得列表有序度变高，最后的插入排序性能大大提高
  - 通过判断列表是否已经有序，提前停止排序过程
  - 通过二叉树结构减少比较次数
  - 以上皆错
- （2 分）以下关于最大堆的描述哪个是正确的？\_\_\_\_\_。
  - 左子树中的关键字都小于右子树中的关键字
  - 左子树中的关键字都小于根节点的关键字
  - 更接近根节点的关键字肯定大于更远离根节点的关键字
  - 更接近根节点的关键字肯定小于更远离根节点的关键字
- （6 分）高度为 3 的 AVL 搜索树，比根节点小的关键字最多有\_\_\_\_\_个，最少有\_\_\_\_\_。



个, 比根节点大的关键字最多有\_\_\_\_\_个, 最少有\_\_\_\_\_个。

- A. 1
- B. 2
- C. 3
- D. 4

5. (4分) 10条边的简单图, 最多有\_\_\_\_\_个顶点, 最少有\_\_\_\_\_个顶点。

- A. 3
- B. 4
- C. 5
- D. 任意多

6. (2分) hash 函数为模运算, 下面哪个值作为模数更好? \_\_\_\_\_。

- A. 1000
- B. 1024
- C. 972
- D. 997

7. (2分) 用数组存储二叉树, 下面哪种结构的二叉树存储空间利用效率最高? \_\_\_\_\_。

- A. 左子树高于右子树
- B. 右子树高于左子树
- C. 完全二叉树
- D. 退化为链的树

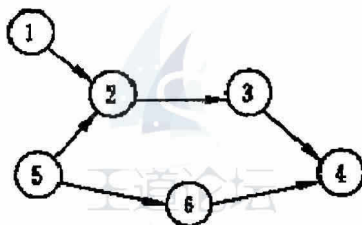
#### 五、简答题 (41 分)

1. (12分) 利用起泡排序将下面字符串序列整理为字典序, 给出每趟起泡后的结果。

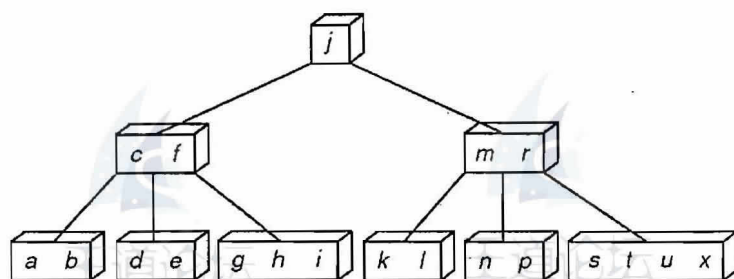
Tim, Dot, Eva, Roy, Tom, Kim, Guy, Amy, Jon, Ann, Jim, Kay, Ron, Jan

2. (16分) 对下面的有向图。

- 1) 给出每个顶点的入度和出度
- 2) 画出邻接链表
- 3) 求所有可能的拓扑序列



3. (8分) 依次删除下图 5 阶 B-树中的关键字 e、f。



4. (5 分) 画出所有由 68、20、84 这 3 个关键字构成的二叉搜索树。

六、(10 分) 设计函数，将给定的数据插入到一棵二叉树中：若树为空，则记录插入根节点，否则，插入较矮的一棵子树（若左右子树等高，插入左子树）