

第一章

- 1.程序的基本要求有哪些?
- 2.面向对象程序设计有哪几个基本要素?
- 3.在 VC++集成环境下,从输入源程序到得到正确的结果,要经过哪些步骤?
- 4.将本章例题中的程序输入到源程序文件 ex1_1.cpp,并在 VC++集成环境下编译、连接和运行。
- 5.在 VC++中,有哪两种注解方法?每一种注解方法适用于什么场合?
- 6.简要说明 C++程序开发的每一个步骤。

第二章

1. 下列选项中,不符合C++语言的语法规则的关键字是 B。// 其实是找正确的关键字
A) integer B) default C) VAR D) cher
2. 下列常量中,不符合C++语言的语法规则的常数是 B。
A) 0xEF B) 1.2e0.6 C) 5L D) '56'
3. 以下选项中,不符合C++语言的基本数据类型是 B。
A) signed short int B) long short C) unsigned long int D) unsigned int
4. 下列表示中,属于C++语言中合法的长整型常数为 C。
A) 652381& B) 326D C) 0L D) 3658921
5. 下列十六进制的整型常数中,不符合C++语法规则的是 C。
A) 0xbe B) 0x2c C) xef D) 0xEF
6. 以下选项中,合法的字符常量是 C。
A) "A" B) 72 C) '\032' D) D
7. C++语言中,基本数据类型是指 C。
A) 整型、浮点型、逻辑型和空类型 B) 整型、浮点型、字符型
C) 整型、浮点型、字符型和空类型 D) 整型、浮点型、字符型和逻辑型
8. C++语言中,运算对象必须是整型的运算符是 C。
A) / B) <= C) %= D) =
9. 设有说明语句: double x,y; 则表达式x=3, y=x+5/3 的值是 C/B。//显示时VC做了处理
A) 4.66667 B) 4 C) 4.0 D) 3
10. 若变量a、i已正确定义,且i已正确赋值,下列合法的赋值表达式是 D。
A) a==1 B) ++i C) a=a++=5 D) a=int(i)
11. 设有说明语句: int a=10,b=9,c=8; 接着顺序执行c=(a=(b-6)); c=(a%8)+(b=5), 则变量a的值是 (1) D, 变量b的值是 (2) C, 变量c的值是 (3) A。
(1) A) 10 B) 3 C) 5 D) 7
(2) A) 6 B) 8 C) 5 D) 9
(3) A) 12 B) 8 C) 5 D) 7
12. 设有说明语句: int a=13,b=9,c; 执行c=a/b+0.8后, c的值为 B。
A) 1.8 B) 1 C) 2.24444 D) 2
13. 若变量a是int类型,并执行了语句a='A'+1.6; 则下列叙述正确的是 D。
A) a 的值是字符 A B) a 的值是浮点型

- C) 不允许字符型和浮点型相加 D) a 的值是字符'A'的 ASCII 值加上 1
14. 变量x、y和z均为double类型且已正确赋值,不能正确表示数学式子 $\frac{x}{y \times z}$ 的C++语言表达式是 B
- A) $x*(1/(y*z))$ B) $x/y*z$ C) $x/y*1/z$ D) $x/y/z$
15. 若a为int类型,且其值为5,则执行表达式 $a=a+a*a$ 后,a的值是 C。
- A) -5 B) 25 C) 0 D) -20
16. 设x、y、z均为int型变量,则执行 $x=y=5; z=++x || ++y$ 后,变量x的值是 (1) C, 变量y的值是 (2) B, 变量z的值是 (3) 1。
- (1) A) 4 B) 5 C) 6 D) 7
 (2) A) 不定值 B) 5 C) 6 D) 7
 (3) A) 10 B) 11 C) 12 D) 13
17. 表达式 $16/4*float(4)+2.0$ 的数据类型是 C。
- A) int B) float C) double D) 不确定
18. 设有说明语句: $int\ m=13,n=3$, 则执行 $m\%=n+2$ 后,n的值是 C。
- A) 5 B) 1 C) 3 D) 0
19. 设a、b、c、d、m、n均为int型变量,且 $a=5、b=6、c=7、d=8、m=2、n=2$, 则逻辑表达式 $(m>a>b)\ \&\&\ (n=c>d)$ 运算后,n的值为 B。//逻辑运算优化的副作用
- A) 3 B) 2 C) 1 D) 0
20. 设有说明语句: $int\ x=8, float\ y=8.8$; 则下列表达式中错误的是 D。
- A) $x\%3+y$ B) $y*y\&\&++x$ C) $(x>y)+(int(y)\%3)$ D) $---x+y$
21. 整型变量m和n的值相等,且为非0值,则以下选项中,结果为零的表达式是 B。
- A) $m | n$ B) $m \wedge n$ C) $m || n$ D) $m \& n$
22. 能正确表示逻辑关系:“ $age \geq 18$ 或 $age \leq 60$ ”的C++语言的表达式是 D。
- A) $age \geq 18$ 或 $age \leq 60$ B) $age \geq 18 | age \leq 60$
 C) $age \geq 18 \&\& age \leq 60$ D) $age \geq 18 || age \leq 60$
23. 下列关于类型转换的描述中,错误的是 D。
- A) 赋值表达式的类型是左值的类型
 B) 逗号表达式的类型是最后一个表达式的类型
 C) 在由低到高精度的类型转换中精度不会降低
 D) 在不同类型对象组成的表达式中,其表达式类型一定是 double 型
24. 设x是int型变量,请写出判断x为奇数的关系表达式 $x\%2 == 1$ 或 $x\%2 != 0$ 或 $x/2*2 == x$ 或 $x\%2$ 。
25. 表示“整数y的绝对值大于8”时值为“真”的C++语言表达式是 $y > 8 || y < -8$ 。
26. 下列表达式的值分别是 0 0.333 1 7。
- A) 1/3 B) 1/3.0 C) 1%3 D) 21/3
27. 设有说明语句: $int\ a=9,b=9,c=9$; 下列表达式的值分别是 4 27 8 162。
- A) $a/=2+b++-c++$ B) $a+=b+c++$ C) $a=-++b-c--$ D) $a*=-b+c--$
28. 下列运算符中, B 的结合性是从左到右。
- A) 单目运算符 B) 关系运算符 C) 三目运算符 D) 赋值运算符
29. 下列表达式的值分别是 1 1 31(1f) 0。// $(a'==(b'<=3))\&5$
- A) $!(5'>8') || 3 < 9$; B) $6 > 3 + 2 - ('0' - 8)$ C) $3 * 5 | 6 < < 2$ D) $'a' == 'b' < = 3 \& 5$
30. 这些语句: $x++; ++x; x=x+1$; 都使变量x中的值加1,请写出一条能使x值加1的赋值语句(不得与列举的相同) $x += 1;$ 。

第三章

1. 设变量ch1,ch2,ch3 为char类型,通过cin>>ch1>>ch2>>ch3; 分别将字符a输入给ch1, 字符b输入给ch2, 字符c输入给ch3, 不正确的输入形式是___B___。

- A) a<换行符> B) a,b,c<换行符>
 b<换行符>
 c<换行符>
C) a<换行符> D) a b<换行符>
 b c<换行符> c<换行符>

2. 下列程序的运行结果是___A___。

```
#include <iostream.h>
void main(void)
{
    cout<<"Zhao:Hello!";      cout<<"How are you?\n";
    cout<<"Liu:I am fine,\t";  cout<<"Thank you\n";
    cout<<6+"\012'<<'t';      cout<<6+"\x12'<<endl;
}

```

- A) Zhao:Hello!How are you?
 Liu:I am fine, Thank you
 16 24
- B) Zhao:Hello!How are you?
 Liu:I am fine, Thank you
 18 18
- C) Zhao:Hello!How are you?
 Liu:I am fine, Thank you
 18 24
- D) Zhao:Hello!How are you?
 Liu:I am fine, Thank you
 16 18

3. 下列程序的运行结果是___D___。

```
#include <iostream.h>
void main(void)
{
    int i=3;      cout<<i+20.5<<" ";
    i=20.5+i;    cout<<i<<endl;
}

```

- A) 23 23 B) 23.5 23.5 C) 23 23.5 D) 23.5 23

4. 下列程序的运行结果是___C___。

```
#include <iostream.h>
void main(void)
{
    int a=9,b=2;
    float x=6.6f,y=1.1f,z;
    z=a/2+b*x/y+1/2;
    cout<<z<<endl;
}

```

- A) 17 B) 15 C) 16 D) 18

5. 下列程序的运行结果是___B___。

```
#include <iostream.h>

```

```
void main(void)
{ int x=178; cout<<oct<<x<<" "<<hex<<x<<endl;}
```

A) 178 178 B) 262 b2 C) 262 262 D) b2 b2

6. 下列程序的运行结果是 A。

```
#include <iostream.h>
void main(void)
{ int x=0; x+=(x=5); cout<<x<<endl;}
```

A) 10 B) 5 C) 0 D) 15

7. 下列程序的运行结果是 D。

```
#include <iostream.h>
void main(void)
{ int a=5,b=4,c=3,d; d=(a>b>c); cout<<d<<endl; }
```

A) 5 B) 3 C) 1 D) 0

8. 下列程序的运行结果是 D。

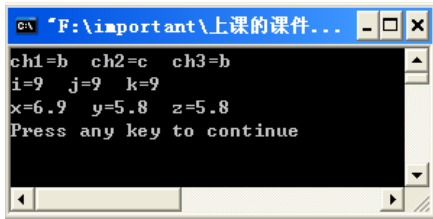
```
#include <iostream.h>
void main(void)
{
    int x=4,y=14,z=15,w;
    w=x+y+z;
    cout<<dec<<"w="<<w<<" ";
    cout<<oct<<"w="<<w<<" ";
    cout<<hex<<"w="<<w<<endl;
}
```

A) w=33 w=33 w=33 B) w=33 w=42 w=22
C) w=34 w=41 w=22 D) w=33 w=41 w=21

9. 下列程序的运行结果是 _____。

```
#include <iostream.h>
void main(void)
{
    char ch1='a',ch2='b',ch3='c';
    int i=9,j=8,k=7;
    double x=3.6,y=5.8,z=6.9;
    ch1=ch2;ch2=ch3;ch3=ch1;
    cout<<"ch1="<<ch1<<" ch2="<<ch2<<" ch3="<<ch3<<endl;
    j=k;k=i;j=k;
    cout<<"i="<<i<<" j="<<j<<" k="<<k<<endl;
    x=y;y=x;x=z;z=y;
    cout<<"x="<<x<<" y="<<y<<" z="<<z<<endl;
}
```

ch1=b ch2=c ch3=b
i=9 j=9 k=9
x=6.9 y=5.8 z=5.8

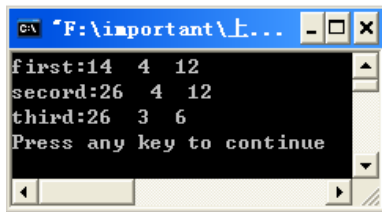


```
ch1=b ch2=c ch3=b
i=9 j=9 k=9
x=6.9 y=5.8 z=5.8
Press any key to continue
```

10. 下列程序的运行结果是_____。

```
#include <iostream.h>
void main(void)
{
    int a=1,b=2,c=3;
    ++a;    c+=++b;
    {
        int b=4,c;
        c=b*3;
        a+=c;
        cout<<"first:"<<a<<" "<<b<<" "<<c<<endl;
        a+=c;
        cout<<"second:"<<a<<" "<<b<<" "<<c<<endl;
    }
    cout<<"third:"<<a<<" "<<b<<" "<<c<<endl;
}
```

first: 14 4 12
second: 26 4 12
third: 26 3 6



```
first:14 4 12
second:26 4 12
third:26 3 6
Press any key to continue
```

11. 给出下列程序的运行结果_____。

```
#include <iostream.h>
#include <iomanip.h>
void main(void)
{
    double e=2.718282828,d;
    cout<<e<<" "<<endl;
    cout<<setiosflags(ios::scientific)<<e<<" "<<endl;
    cout<<setiosflags(ios::fixed)<<e<<" "<<endl;
    d=12.0/3;
    cout<<d<<endl;
}
```

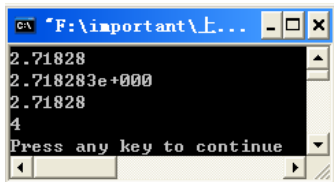
```
}
```

2.71828

2.718283e+000

2.71828

4



```
2.71828
2.718283e+000
2.71828
4
Press any key to continue
```

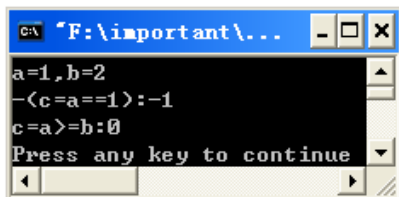
12. 给出下列程序的运行结果_____。

```
#include <iostream.h>
void main(void)
{
    int a=1,b=2,c;
    cout<<"a=1,b=2"<<endl;
    cout<<"-(c==1):"<<-(c==1)<<endl;
    cout<<"c>=b:"<<(c>=b)<<endl;
}
```

a=1 b=2

-(c==1): -1

c>=b: 0



```
a=1, b=2
-(c==1):-1
c>=b:0
Press any key to continue
```

13. 根据题目要求，编写完整的程序。

(1) 从键盘上输入一个整数 x 后，将其低六位全部置 1，其余各位不变。然后，用十进制和十六进制输出 x 的值。

```
#include <iostream.h>
void main()
{
    int x;
    cout<<"输入一个整数: ";
    cin>>x;
    x = x | 63; // 或 x = x | 0x3f; 或 x |= 0x3f;
    cout<<"十进制输出整数: "<<x<<endl;
    cout<<"十六进制输出整数: "<<hex<<x<<endl;
}
```

(2) 从键盘上输入一个整数 x 后，将其高六位全部置 0，其余各位不变。然后，用十进制和十六进制输

出 x 的值。

```
#include <iostream.h>
void main()
{
    int x;
    cout<<"输入一个整数: ";
    cin>>x;
    x = x & 67108863; // 或 x&=0x3ff;
    cout<<"十进制输出整数: "<<x<<endl;
    cout<<"十六进制输出整数: "<<hex<<x<<endl;
}
```

(3) 从键盘上输入任意一个摄氏温度 C，求该温度对应的华氏温度 F ($F=5/9(c-32)$) 及绝对温度 T ($273.15+C$)。

```
#include <iostream.h>
void main()
{
    float c,f,t;
    cout<<"摄氏温度: ";
    cin>>c;
    f = c*9/5 +32;
    t = 273.15 + c;
    cout<<"华氏温度: "<<f<<endl;
    cout<<"绝对温度: "<<t<<endl;
}
```

(4) 从键盘上输入任意一个字符，求出它的 ASCII 码值并输出。

```
#include <iostream.h>
void main()
{
    char c;
    cout<<"输入任意一个字符: ";
    cin>>c;
    cout<<"它的 ASCII 码值是: "<<c + 0<<endl; //或 int(c)
}
```

(5) 从键盘上输入任意两个实数，求它们的和、差与积，并输出。

```
#include <iostream.h>
void main()
{
    double r1,r2;
    cout<<"输入任意两个实数: ";
    cin>>r1>>r2;
    cout<<"它们的和: "<<r1 + r2<<endl;
    cout<<"它们的差: "<<r1 - r2<<endl;
    cout<<"它们的积: "<<r1 * r2<<endl;
}
```

(6) 输入任意一个实数，将其整数部分与小数部分分开，并分别输出在两行上。例如，实数 32.68，输出：

32

0.68

```
#include <iostream.h>
void main()
{
    double trans;
    cout<<"输入要转换的实数: "<<endl;
    cin>>trans;
    cout<<(int)trans<<endl;
    cout<<trans - (int)trans<<endl;
}
```

或者：

```
#include <iostream.h>
void main()
{
    float x,b;
    int a;
    cout<<"x=" ;
    cin>>x;
    a=x; b=x-a; // 仅仅是编译时警告有可能会丢失精度
    cout<<"a"<<a<<endl<<"b="<<b<<endl;
}
```

(7) 从键盘上输入三角形的三边长（均为实数），求三角形的面积和周长，并输出计算结果。

```
#include <iostream.h>
#include <math.h>
void main()
{
    float a,b,c,s,area;
    cout<<"a,b,c=" ;
    cin>>a>>b>>c;
    if ((a+b>c)&&(b+c>a)&&(c+a>b)){
        s=(a+b+c)/2;
        area=sqrt(s*(s-a)*(s-b)*(s-c));
        cout<<"s="<<s<<endl<<"area="<<area<<endl;
    }
    else cout<<"Error!"<<endl;
}
```


第四章

1. C++语言程序设计的三种基本结构是 A。
A) 顺序结构、选择结构、循环结构 B) 循环结构、转移结构、顺序结构
C) 递归结构、循环结构、转移结构 D) 嵌套结构、递归结构、顺序结构
2. 为了避免嵌套的条件语句if-else的二义性，C++语言规定else与 B 配对。
A) 缩排位置相同的 if B) 在其之前未配对的最近的 if
C) 在其之后最近的 if D) 同一行上的 if
3. 下列关于条件语句的描述中，错误的是 D。
A) 一个 if 只能有一个 else 与之配对 B) if 语句中可以包含循环语句
C) if 语句中可以有多个 else if 子句 D) if 语句中不能包含 switch 语句
4. 下列关于switch语句的描述中，正确的是 D。
A) switch 语句中 default 子句只能放在最后
B) switch 语句中 case 后边的表达式可以是实型表达式
C) switch 语句中，每个语句序列中必须有 break 语句
D) 在 switch 语句中可以有 default 子句，也可以没有
5. 下列关于for循环的描述中，正确的是 B。
A) for 循环只能用于循环次数确定的情况 B) for 循环语句中，可以包含其它循环语句
C) for 循环是先执行循环体语句，后测试条件 D) 在 for 循环中，不能用 goto 语句跳出循环体
6. 下列关于break语句的描述中，不正确的是 B。
A) break 语句可用于循环体内，终止循环的执行
B) break 语句可用于 if 体内，终止 if 语句的执行
C) break 语句可用于 switch 体内，终止 switch 语句的执行
D) 在一个循环体内，break 语句可以出现多次。
7. 下列for语句的循环次数是 D 次。
for (int i=1;i<=5;sum++) sum+=i;
A) 5 B) 4 C) 0 D) 无限
8. 下列for语句的循环次数是 A 次。
for(int k=0; ;k++)
A) 无限 B) 0 C) 有语法错，不能执行 D) 1
9. 以下程序段的输出结果是 C。
int x=3;
do{
 cout<<(x-=2)<<" ";
}while (!(--x));
A) 1 B) 3 0 C) 1 -2 D) 死循环
10. 若a、b、c1、c2、x、y、均是整型变量，正确的switch语句是 B。
A) switch(a+b); {
 case 1: y=a+b; break;
 case 0: y=a-b; break;
} case 3:y=b-a;break;
B) switch(a*a+b*b) {
 case 3:
 case 1:y=a+b;break;
}
C) switch a {
 case c1 :y=a-b; break;
D) switch(a-b) {
 default:y=a*b;break;

```

    case c2: x=a*d; break ;
    default: x=a+b;
}

```

```

    case 3:case 4:x=a+b;break;
    case c1+10: y=a-b;break;
}

```

11. 下列程序的输出结果是 C。

```

#include <iostream.h>
void main(void)
{
    int x=1,i=1;
    for (;x<50;i++){
        if(x>=10)break;
        if(x%2!=0){
            x+=3; continue;
        }
        x--1;
    }
    cout<<x<<' '<<i<<endl;
}

```

A) 12 7

B) 11 6

C) 12 6

D) 11 7

12. 下列程序的输出结果是 C。

```

#include <iostream.h>
void main(void)
{
    int n='m';
    switch(n++){
        default: cout<<"error";break;
        case 'k':case 'K':case 'l':case 'L':cout<<"good"<<endl;break;
        case 'm':case 'M':cout<<"pass"<<'\t';
        case 'n':case 'N':cout<<"warn"<<endl;
    }
}

```

A) pass

B) warn

C) pass warn

D) error

13. 编写完整的程序。

(1) 任意输入两个数，完成这两个数的四则运算。

就是完成+、-、×、/ 运算

```

#include <iostream.h>
#include <stdlib.h>
void main()
{
    int a,b,c;
    char ch;
    cout<<"请输入计算式：（如： 3+5）";
    cin>>a>>ch>>b;
    switch(ch) {
        case '+': c=a+b; break;

```

```

        case '-': c=a-b; break;
        case '*': c=a*b; break;
        case '/':
            if (b==0) { cout<<"除数为零!"<<endl; exit(1); }
            c=a/b; break;
        default:
            cout<<"运算符错!"<<endl; exit(2);
    }
    cout<<"="<<c<<endl;
}

```

(2) 输入 1~7 之间的数，按英文名称输出是星期几。

利用 switch 语句，注意 break，若输入值不在范围内的处理。

(3) 输入若干个字符，统计输入的数字字符的个数。

数字字符的 ASCII 码范围：48~57（'0'~'9'）

```

#include <iostream.h>
void main()
{
    int count = 0;
    char ch;
    do {
        ch=cin.get();
        if (ch>='0' && ch<='9') count++;
    }while(ch!='\n');
    cout<<count<<endl;
}

```

(4) 求一元二次方程 ax^2+bx+c 的根。任意输入系数a、b、c的值，求出方程的两个根。

```

#include <iostream.h>
#include <math.h>
void main()
{
    float a,b,c,d;
    cout<<"请输入方程的系数 a,b,c: ";
    cout<<"a = ";
    cin>>a;
    cout<<"b = ";
    cin>>b;
    cout<<"c = ";
    cin>>c;
    d = b * b - 4 * a * c;
    if(a == 0)
        cout<<"不是二次方程!"<<endl;
}

```

```

else if(d == 0)
    cout<<"一个实根: "<< -b/(2 * a) << endl;
else if(d > 0)
    cout<<"两个实根: "<< (-b + sqrt(d))/(2 * a) << ", "<< (-b - sqrt(d))/(2 * a) << endl;
else
    cout<<"无实根! " << endl;
}

```

(5) 已知数列的前 5 项为: $\frac{1}{2}, \frac{3}{2}, \frac{5}{3}, \frac{8}{5}, \frac{13}{8}$, 按照这种规律求出该数列的前 20 项。

```

#include <iostream.h>
void main()
{
    int i,m,n,k;
    float s = 0;
    m = 1, n = 2;
    for(i = 1; i <= 20; i++)
    {
        cout<<"n/m = " << n << "/" << m << endl;
        s = s + 1.0 * n/m;
        k = m;
        m = n;
        n = k+n;
    }
    cout<<"s = " << s << endl;
}

```

或者

```

#include <iostream.h>
void main()
{
    int a=2,b=1,c;
    for (int i=1;i<=20;i++){
        cout<<a<<"/"<<b<<endl;
        c=a+b;
        b=a;
        a=c;
    }
}

```

(6) 求出 1~100 之间的完全数。所谓完全数是指该数刚好等于它的因子之和 (自己本身除外)。例如, 6 的因子为 1, 2, 3, 且 $6=1+2+3$, 因此, 6 是一个完全数。

```

#include <iostream.h>
void main()
{

```

```

int a,i,m;
for(a = 1; a <= 100; a++)
{
    for(m = 0, i = 1; i <= a/2; i++)
        if(!(a%i)) m += i;
    if(m == a) cout<<a<<endl;
}

```

或者

```

#include <iostream.h>
void main()
{
    int k;
    for (int i=2;i<=100;i++){
        k=0;
        for (int j=1;j<i;j++)
            if (i%j==0) k+=j;
        if (k==i) cout<<i<<endl;
    }
}

```

(7) 输入任意一个正偶数，将其分解为两个素数之和。例如， $8=3+5$ 。

```

#include <iostream.h>
#include <math.h>
void main()
{
    int a,b,c,d;
    cout<<"a=";
    cin>>a;
    if(a <= 0 || a%2 != 0)
        cout<<"输入一个正偶数!"<<endl;
    else
        for(b = 3; b <= a/2; b += 2)
        {
            for(c = 2; c <= sqrt(b); c++)
                if(b % c == 0) break;
            if(c > sqrt(b)) d = a - b;
            else break;
            for( c = 2; c <= sqrt(d); c++)
                if(d % c == 0) break;
            if(c > sqrt(d)) cout<<b<<"+"<<d<<"="<<a<<endl;
        }
}

```

或者使用函数定义来实现：

```

#include <iostream.h>
#include <math.h>
#include <stdlib.h>

int p(int n)
{
    if (n<=1) return 0;
    for (int i=2;i<=sqrt(n);i++)
        if (n%i==0) { return 0; }
    return 1;
}

void main()
{
    int k;
    cout<<"k=";
    cin >>k;
    if (k%2) { cout<<"error!"<<endl; exit(2); }
    for (int i=2;i<=k/2;i++){
        if (p(i)&& p(k-i)) cout<<i<<'+'<<k-i<<'='<<k<<endl;
    }
}

```

(8) 输出下列图形。

```

      a
     b b
    c c c
   d d d d
  e e e e e
 f f f f f f
g g g g g g g

```

参考《C++语言程序设计习题集》

```

#include <iostream.h>
void main()
{
    int i,j;
    char display = 'a';
    for(i = 1; i <= 7; i++)
    {
        for(j = 1; j <= (14 - 2 * i); j++)
            cout<<' ';
        for(j = 1; j <= i; j++)
            cout<<(char)(display + i - 1)<<" ";
        cout<<endl;
    }
}

```

```
}
```

或者

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

```
void main()
```

```
{
```

```
    char c='a';
```

```
    int n=9;
```

```
    for (int i=2;i<=n;i++,c++){
```

```
        for (int j=1;j<n-i+1;j++) cout<<' ';
```

```
        for (int j=1;j<i;j++) cout<<c;
```

```
        cout<<endl;
```

```
    }
```

```
}
```

第五章

1. 对于C++的函数，正确的叙述是 A。
A) 函数的定义不能嵌套,但函数的调用可以嵌套 B) 函数的定义可以嵌套,但函数的调用不能嵌套
C) 函数的定义和调用都不能嵌套 D) 函数的定义和调用均可以嵌套
2. 对于C++的函数，描述正确的是 B。
A) 一个函数中可定义另一个函数 B) 函数的定义是平行的、平等的
C) 函数不能自己调用自己 D) 函数定义是不平等的，分主次的
3. 假设A和B是两个函数，正确的叙述是 D。
A) 函数 A 中调用 B，则 B 中不能调用函数 A B) 函数 A 和 B 中都不能调用自己
C) 若函数 A 中调用 B，则 A 中不能调用自己 D) 函数 A 调用 B 时，函数 B 中可调用 A
4. 下列程序执行的结果是 C。

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

```
int f(int Int)
```

```
{
```

```
    if (Int==0) return 1;
```

```
    return (Int+f(Int-1));
```

```
};
```

```
void main(void)
```

```
{
```

```
    int inT=9;
```

```
    cout<< "result=" << f(inT) << "\n";
```

```
}
```

- A) result=1 B) result=37 C) result=46 D) 编译错
5. 下列递归函数中，均能正确结束递归的函数定义是 C。 //如果n是负数则D不能结束
A) int f(int n) { if (n<1) return 1; else return n*f(n+1); }
B) int f(int n) { if (n>1) return 1; else return n*f(n-1); }
C) int f(int n) { if (abs(n)<1) return 1; else return n*f(n/2); }
D) int f(int n) { if (n>1) return 1; else return n*f(n*2); }

6. 有关函数重载的正确说法是 B。

- A) 函数名不同, 但参数的个数和类型相同 B) 函数名相同, 参数的个数或参数的类型不同
C) 函数名相同, 参数的个数和类型也相同 D) 函数名相同, 返回值不同, 与函数的参数无关

7. 满足C++函数重载规则的是 D。

- A) float fun(float x);void fun(float y); B) float funa(float x);void fun(float x, float y);
C) float fun(float y); void fun(float x); D) float fun(float x, float y); void fun(float y);

8. 下面程序的运行结果是 D。

```
#include <iostream.h>
#define AA      10
#define D(x)   x * x                               //E
void main(void)
{
    int x=1,y=2,t;
    t=D(x+y)*AA;                                   //F
    cout<<t;
}
```

- A) F 行中的表达式有错 B) 60 C) 50 D) E 行中的宏定义有错

9. 设有以下宏定义和语句, 则i的值为 B。

```
#define ONE    1
#define TWO    (ONE+ONE)
#define THREE  ONE+TWO
...
```

i=THREE*3+TWO*2;

- A) 13 B) 11 C) 9 D) 8

10. 执行以下程序后, 输出是 D。

```
#include <iostream.h>
#define P      5
#define R      2+P
void main(void)
{ float a1;a1=P* R * R;cout << "a1="<<a1<<"\n"; }
```

- A) a1=75 B) a1=245 C) a1=49 D) a1=25

11. 执行以下程序后, 输出是 A。

```
#include <iostream.h>
static int c;
void main(void)
{
    #if c*3
        int i=10.88; cout<<"i="<<i<<"\n";
    #else
        int j=10000.99;cout<<"j="<<j<<"\n";
    #endif
}
```

- A) j=10000 B) j=10.88 C) 语法错, 无输出 D) j=10

12. 有关编译预处理命令的正确说明是 D。

- A) 以"#"开头, 必放在在程序开头
 B) 以#开头的行, 后面不可加分号
 C) 以"#"开头, 可出现在一行中的任何位置
 D) 以"#"开头的行, 可出现程序中的任一位置

13. 下面程序的运行结果是__C__。

```
#include <iostream.h>
int f(int x)
{ static int u=1; x+=x; return u*=x; }
void main(void)
{ int x=10; cout<<f(x)<< '\t'; cout<<f(x)<<endl; }
```

- A) 10 20 B) 20 800 C) 20 400 D) 20 20

14. 下面程序的运行结果是__A__。

```
#include <iostream.h>
int t(void )
{ static int i =1; i+=2; return i; }
int t1(void )
{ int j =1; j+=2; return j; }
void main(void)
{
  int j=2;
  t(); cout << "I="<<t()<<" ";
  t1(); cout << "J="<<t1()<< "\n";
}
```

- A) I=5 J=3 B) I=5 J=5 C) I=3 J=5 D) I=3 J=-3

15. 下面程序的运行结果是__A__。

```
#include <iostream.h>
float p(float x,int i)
{ x=x+2.5; i=i+x;return x; }
void main(void)
{
  int i=10;float x=3.25;
  x=p(x,i)-1;cout<<"x="<<x <<" ,i="<<i;
}
```

- A) x=4.75,i=10 B) x=5.75,i=12 C) x=5.75,i=10 D) x=4.75,i=12.5

16. 下面程序的运行结果是__B__。

```
#include <iostream.h>
int loop( int n)
{
  if(n==1) return 10;
  else
    if(n%2==0) return loop(n-1)+2;
    else return loop(n-1)+3;
}
void main(void)
{ cout<<loop(3)<<endl; }
```

- A) 14 B) 15 C) 16 D) 10

17. 读程序，写出以下程序的输出结果： 10 24。

```
#include <iostream.h>
int c_multiple(int a,int b)
{
    int i;
    for (i=(a>b?a:b);i<=a*b;i++)if (i%a==0&&i%b==0) return i;
}
void main(void)
{
    cout <<c_multiple(2,5)<<"\n";
    cout <<c_multiple(6,8)<<"\n";
}
```

18. 读程序，写出以下程序的输出结果：

9
7
5

。

```
#include <iostream.h>
void f(int n)
{
    if(n/10){ cout<<n%10<<"\n"; f(n/10); }
    else cout<<n;
}
void main(void)
{ f(579); cout<<endl; }
```

19. 读程序，写出以下程序的输出结果：

34
3
0

。

```
#include <iostream.h>
void f(int n)
{
    cout<<n/10<<"\n";
    if(n/10) f(n/10);
}
void main(void)
{ f(345); cout<<endl; }
```

20. 读程序，写出以下程序的输出结果：

20	100
10	100

```
#include <iostream.h>
int x=100;
void main(void)
{
    int x=10,k=20;
    { int x=20;
      k::x;
      cout<<x<<"\t"<<k<<endl;
    }
    cout<<x<<"\t"<<k<<endl;
}
```

21. 读程序，写出以下程序的输出结果：

x=20
y=40
x=20
y=40

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

```

void swap(int p1,int p2)
{   int p; p=p1;p1=p2;p2=p;   }
void main(void)
{
    int x=20,y=40;
    cout<<"x="<<x<<"\n"<<"y="<<y<<"\n";
    swap(x,y);
    cout<<"x="<<x<<"\n"<<"y="<<y<<"\n";
}

```

22. 根据题目要求，设计出完整的程序。

(1)设计一个程序，要求输入三个整数，求出其中的最大数并输出。必须用函数实现求两个数中的大数。

```

#include <iostream.h>
int max(int x,int y)
{
    return (x>y?x:y);
}
void main(void)
{
    int a,b,c;
    cout <<"输入三个整数： ";
    cin >>a>>b>>c;
    cout <<"最大数为： "<<max(max(a,b),c)<<"\n";
}

```

(2)设计一个程序，计算组合数： $C(m, r) = m! / (r! \times (m-r)!)$ ，其中 m, r 为正整数，且 $m > r$ 。分别求出 $C(4, 2)$ ， $C(6, 4)$ ， $C(8, 7)$ 的组合数。求阶乘和组合数分别用函数来实现。

```

#include <iostream.h>
int fac(int n)
{
    int f=1;
    for (int i=1;i<=n;i++) f*=i;
    return f;
}
int c(int m,int r)
{
    if (m<0||r<0||m<r) return 0;
    else return (fac(m)/fac(r)/fac(m-r));
}
void main(void)
{
    int n,m,c1;
    cout <<"输入 n,m 的值 (n>m>0): ";
    cin >>n>>m;
    c1=c(n,m);
}

```

```

if (c1==0) cout <<"输入数据错误! \n";
    else cout <<"C("<<n<<","<<m<<")="<<c1<<"\n";
}

```

(3)设计一个程序，将输入的十进制整数转换为相应的十六进制数并输出。设计一个函数实现数制转换。

```

#include <iostream.h>
void trans(int, int);
void main()
{
    int n;
    cout<<"输入一个整数: ";
    cin>>n;
    trans(n, 2);
}
void trans(int n, int base)
{
    int num[20], i = 0, j, m = n;
    do
    {
        i++;
        num[i] = m % base;
        m = m/base;
    }while(m != 0);
    cout<<n<<"转换成"<<base<<"进制的结果为: ";
    for(j = i; j >= 1; j--)
        if(num[j] <= 9)
            cout<<num[j];
        else
            cout<<(char)(num[j] - 10 + 'a');
    cout<<endl;
}

```

(4)设计一个程序，求出 5~500 之间的所有素数，要求每行输出 5 个素数。设计一个函数来判断一个整数是否为素数。

```

#include <iostream.h>
#include <math.h>
int prime(int n)
{
    int j;
    if (n<=1) return 0;
    j=sqrt(n);
    for (int k=2;k<=j;k++)
        if (n%k==0) return 0;
    return 1;
}

```

```

void main(void)
{
    int k=0;
    for (int j=5;j<100;j+=2)
        if (prime(j)) {
            cout <<j<<"\t";
            if (++k%5==0) cout <<"\n";
        }
    cout <<"\n";
}

```

【2.71】 参考答案:

```

#include <math.h>
bool Fun(int prime)
{
    int i = 2;
    bool flag = true;
    while (i <= sqrt(prime) && flag)
    {
        if (prime % i == 0)
            flag = false;
        else
            i++;
    }
    return flag;
}

```

注释：判断一个自然数是否是素数，只需要依次用 $2 \sim \sqrt{n}$ 做除数去除 n ，若 n 不能被其中任何一个数整除，则 n 是素数。

(5)设计一个程序，输入两个正整数，求出这两个整数的最小公倍数。求两个数的最小公倍数用一个函数来实现。

```

#include <iostream.h>
int handle(int x, int y)
{
    int s = x;
    while(1)
    {
        if(s%y == 0) break;
        s += x;
    }
    return s;
}
void main()
{
    int x,y,s;

```

```

cout<<"输入两个正整数: ";
cin>>x>>y;
s = handle(x, y);
cout<<"最小公倍数: "<<s<<endl;
}

```

(6)设计一个程序，输入两个正整数，求出这两个整数的最大公约数。求两个数的最大公约数用一个函数来实现。辗转相除法： $r=a\%b$ ；若 $r=0$ 则输出 b ；否则 $a=b$ ； $b=r$ ；

```

#include <iostream.h>
int handle(int i, int j)
{
    int r = 0;
    if(i<j)
    {
        r = i;
        i = j;
        j = r;
    }
    r = i % j;
    while(r)
    {
        i = j;
        j = r;
        r = i % j;
    }
    return j;
}

void main()
{
    int r,i,j;
    cout<<"输入第一个正整数: "<<"i = ";
    cin>>i;
    cout<<"输入第二个正整数: "<<"j = ";
    cin>>j;
    r = handle(i,j);
    cout<<"最大公约数: "<<r<<endl;
}

```

或者

```

#include <iostream.h>
int c_devisor(int a,int b)
{
    int r;
    if (b==0) return 0;
}

```

```

    r=a%b;
    while (r!=0){
        a=b; b=r; r=a%b;
    }
    return b;
}
void main(void)
{
    int a,b;
    cout <<"输入两个整数: ";
    cin >>a>>b;
    cout <<"最大公约数为: "<<c_devisor(a,b)<<"\n";
}

```

或者

```

#include <iostream.h>
int c_devisor(int a,int b)
{
    int i;
    for (i=(a<b?a:b);i>=1;i--){
        if (a%i==0&&b%i==0) return i;
    }
}
void main(void)
{
    int a,b;
    cout <<"输入两个整数: ";
    cin >>a>>b;
    cout <<"最大公约数为: "<<c_devisor(a,b)<<"\n";
}

```

(7)设计一个程序，通过重载求两个数中大数的函数 max，分别实现求两个整数、两个实数和两个双精度数的大数。

```

int max(int, int)
float max(float, float)
double max(double, double)

//参考下面的代码
#include "iostream.h"
int max2(int,int);
int max3(int , int , int);
void main()
{
    int x , y, z , t;
    cout<<"请输入一个整数: ";

```

```

cin>>x;
cout<<"请输入一个整数: ";
cin>>y;
cout<<"请输入一个整数: ";
cin>>z;
t=max3(x , y , z);
cout<<endl<<" max="<<t;
}
int max2(int x1 ,int y1 )
{ if (x1>y1)
    return x1;
  else
    return y1;
}
int max3(int x1 ,int y1, int z1)
{
  int t;
  t=max2(x1,y1);
  t=max2(t,z1);
  return t;
}

```

(8)设计一个程序，求出输入三个实数中的最大数，并输出计算结果。要求用内联函数实现求三个实数中的最大数。

`inline`（定义内联函数只是告诉编译器用函数体代替该函数的调用。）

`(a>b?a:b)>c?(a>b?a:b):c`

(9)定义一个求 $n!$ 的函数，其中 n 为正整数，其缺省值为 10。

//将下行中的 `int Fun(int n)` 改为 `int Fun(int n = 10)`，递归函数实现

```

int Fun(int n)
{
    if (n == 1)
        return 1;
    else
        return n * Fun(n - 1);
}

```

或者

```

int fac(int n=10)
{
    int f=1;
    for (int i=1;i<=n;i++) f*=i;
    return f;
}

```

(10)设计一个参数个数可变的函数，第一个参数 n 为参数的个数。求出这 n 个参数的平均值并返回平均值。

(11)设计一个参数个数可变的函数，第一个参数 n 为参数的个数。调用这函数时，若 n 的值为 3，则求出后三个参数之积并返回该值；若 n 的值为 4，则求出后四个参数的平均值并返回该值。

(12)用递归函数实现求 Fibonacci 数列的前 n 项， n 作为函数的参数。

【2.79】斐波那契 (Fibonacci) 数列的前 10 项为 1, 1, 2, 3, 5, 8, 13, 21, 34, 55。试求解斐波那契 (Fibonacci) 数列的第 n 项。

【2.79】参考答案：

① 用递归方法实现 Fibonacci 数列，递归方程是：

$$Fibonacci(n) = \begin{cases} 1 & (n=1) \\ 1 & (n=2) \\ Fibonacci(n-1) + Fibonacci(n-2) & (n \geq 3) \end{cases}$$

```
int Fibonacci(int n)
{
    if ((n == 1) || (n == 2))
        return 1;
    else
        return Fibonacci(n - 1) + Fibonacci(n - 2);
}
```

② 用非递归方法实现 Fibonacci 数列

```
int Fibonacci(int n)
{
    int Fib[100];
    Fib[1] = 1;
    Fib[2] = 1;
    for (int i = 3; i <= n; i++)
        Fib[i] = Fib[i-1] + Fib[i-2];
    return Fib[n];
}
```

注释：数组 Fib 用以存放 Fibonacci 数列的前两项，数组大小应该保证能取到 Fibonacci 数列的第 n 项。

(13)当 $x > 1$ 时，Hermite 多项式定义为：

$$H_n(x) = \begin{cases} 1 & n = 0 \\ 2x & n = 1 \\ 2xH_{n-1}(x) - 2(n-1)H_{n-2}(x) & n > 1 \end{cases}$$

当输入实数 x 和整数 n 后，求出 Hermite 多项式的前 n 项的值。分别用递归函数和非递归函数实现。

```
#include <iostream.h>
float H(int n,float x)
{
    if (n==0) return (1);
    if (n==1) return (2*x);
    return 2*x*H(n-1,x)-2*(n-1)*H(n-2,x);
}
void main(void)
{
    int n;
```

```

float x;
cout <<"输入一个整数和一个实数： ";
cin >>n>>x;
if (x<=0||n<0) cout <<"输入数据错!";
else {
    for (int m=0;m<n;m++){
        cout <<H(m,x)<<"\t";
        if (m%4==0) cout <<"\n";
    }
}
}

```

(14)阿克曼函数定义为:

$$Acm(m,n) = \begin{cases} n+1 & m=0 \\ Acm(m-1,1) & n=0 \\ Acm(m-1,Acm(m,n-1)) & n>0, m>0 \end{cases}$$

其中 m, n 为正整数。设计一个程序，分别求出 $Acm(5, 3)$, $Acm(4, 2)$ 和 $Acm(5, 5)$ 的值。

要求用一个函数实现求 $Acm(m, n)$ 的值。

```

long Fun(int m, int n)
{
    int temp = 0;
    if(m<0||n<0)exit(0);
    if(m == 0)
        return n + 1;
    else if( n == 0)
        return Fun(m-1,1)
    else
    {
        temp = Fun(m,n-1);
        return Fun(m-1,temp);
    }
}

```

完整的可运行的程序

```

#include <iostream.h>
int acm(int m,int n)
{
    if (m==0) return (n+1);
    if (n==0) return acm(m-1,1);
    return acm(m-1,acm(m,n-1));
}
void main(void)
{

```

```

int n,m;
for (m=0;m<4;m++){
    for (n=0;n<9;n++) cout <<acm(m,n)<<"\t";
    cout <<"\n";
}
}

```

(15)设计一个程序，把输入的整数实现逐位正序和反序输出。如输入一个整数 3456，则输出 3456 和 6543。分别设计两个函数，一个实现正序输出；另一个实现反序输出。算法提示：重复除以 10 求余，直到商为 0 为止。如 $3456\%10$ 的余数为 6，商为 345。 $345\%10$ 的余数为 5，商为 34。 $34\%10$ 的余数为 4，商为 3。 $3\%10$ 的余数为 3，商为 0。至此结束。在函数内先输出余数，后递归，则为反序输出；若先递归后输出余数，则为正序输出。

【1.109】 参考答案：

```

#include<iostream.h>
void main()
{
    int Num, Digital;
    cout << "Please input a number: ";
    cin >> Num;
    do
    {
        Digital = Num%10;           //①
        Num /= 10;                  //②
        cout << Digital;
    }while(Num>0);
}

```

注释：本题的要点在于如何一位一位地读取一个整数中的各位数字。可以采用如下方法：先取出该整数的个位数，即语句①；然后再取该整数除个位以外的其余部分，即语句②。

或下面另一种实现

```

#include <iostream.h>
void order(int n)
{
    if (n==0) return;
    order(n/10);
    cout <<n%10;
}
void reverse(int n)
{
    if (n==0) { cout <<"\n"; return; }
    cout <<n%10;
    reverse(n/10);
}

```

```

}
void main(void)
{
    int n;
    cout <<"输入一个正整数: ";
    cin >>n;
    cout <<"正序: "
    order(n);
    cout <<"\n 反序: "
    reverse(n);
}

```

(16)设计一个程序，求出七个数中的最大数。要求将求三个实数的最大数的函数放在一个头文件 myfun.h，在源程序文件 pro.cpp 中包含这个头文件，并实现输入七个实数，通过调用函数实现求最大值。

```
include myfun.h
```

在 pro.cpp 中写入上述语句，就可以调用 myfun.h 中的函数了。

(17)定义带参数的宏 MAX (A, B) 和 MIN (A, B)，分别求出两数中的大数和小数。在主函数中输入三个数，求出这三个数中的最大数和最小数。

```
MAX(A, B)  A>=B?A:B
```

```
MIN(A, B)  A<=B?A:B
```

(18)已知三角形的三条边 a, b, c，则三角形的面积为：

$$area = \sqrt{s(s-a)(s-b)(s-c)}$$

其中 $s = (a+b+c) / 2$ 。设计一个程序，分别用带参宏和函数求三角形的面积。

```
define S(a,b,c) ((a + b + c)/2)
```

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

```
#include <math.h>
```

```
//#define s(a,b,c) (a+b+c)/2
```

```
//#define area(a,b,c) sqrt(s(a,b,c)*(s(a,b,c)-a)*(s(a,b,c)-b)*(s(a,b,c)-c))
```

```
#define s (a+b+c)/2
```

```
#define area(a,b,c) sqrt(s*(s-a)*(s-b)*(s-c))
```

```
void main(void)
```

```
{
```

```
    float a,b,c;
```

```
    cout <<"输入三角形的三条边: ";
```

```
    cin >>a>>b>>c;
```

```
    if (a+b>c && b+c>a && c+a>b)
```

```
        cout <<area(a,b,c)<<"\n";
```

```
    else
```

```
        cout <<"输入数据错! "<<"\n";
```

```
}
```

(19)设计一个程序，分别求出输入的十个数的积和累加和，使用条件编译输出调试信息（中间结果）。

第六章

1. 设有说明语句：int a[10]={9,1,6,0,8}; 数组元素a[3]的值是 D。
A) 9 B) 1 C) 6 D) 0
2. 以下数组说明语句中，错误的是 D。
A) int a[]={1,2}; B) char a[3]; C) char s[10]="test"; D) int n=5,a[n];
3. 以下数组说明中，不正确的是 B。
A) int b[][3]={0,1,2,3}; B) int d[3][]={{1,2},{1,2,3},{1,2,3,4}};
C) int c[100][100]={0}; D) int a[2][3];
4. 设有说明语句：int b[][3]={{9},{1,6},{0,8},{1,2,3}}; 数组元素b[3][2]的值是 A。
A) 3 B) 8 C) 6 D) 9
5. 下列字符数组说明中，正确的是 B。
A) char s4[2][3]={"xyz","abc"}; B) char s1[]="xyz";
C) char s3[][]={'x','y','z'}; D) char s2[3]="xyz";
6. 下列选项中，不正确的是 D。
A) char s4[]="Ctest\n"; B) char s3[20]="Ctest";
C) char s2[]={'C','t','e','s','t'}; D) char s1[10];s1="Ctest";
7. 设有说明语句：char s[5][5]={"abc","efgh"}; 值为字符g的数组元素是 D。
A) s[4][4] B) s[1][4] C) s[1][3] D) s[1][2]
8. 设有说明语句：char s[10]; 对s的数组元素正确引用是 C。
A) s(6) B) s[10] C) s[6+1] D) s[1.5]
9. 执行以下程序段后，t的值是 A。

```
int b[3][3]={0,1,2,0,1,2,0,1,2},i,j,t=1;
for(i=0;i<3;i++)
    for(j=i;j<=i;j++) t=t+b[i][i]+b[j][j];
```


A) 7 B) 9 C) 4 D) 3
10. 当调用函数时，实参是一个数组名，其作用是 C。
A) 传递数组的长度 B) 传递数组中的一个元素
C) 使形参数组与实参数组共用同一空间 D) 错误的，数组名不能作为实参
11. 下列程序的输出结果是 A。

```
#include <iostream.h>
void main(void)
{
    int i, k, a[10], p[3];
    k=5;
    for (i=0;i<10;i++) a[i]=i;
    for (i=0;i<3;i++) p[i]=a[i]*(i+1);
    for (i=0;i<3;i++) k+=p[i]*2;
    cout<<k<<<endl;
}
```


A) 21 B) 22 C) 23 D) 24
13. 下列程序的输出结果是 D。

```

#include <iostream.h>
void main(void)
{
    char w[][10]={ "ABCD","EFGH","IJKL","MNOP"},k;
    for(k=1;k<3;k++) cout<<w[k];
    cout<<endl;
}

```

A) BCD B) EFGH C) IJKL D) EFGH IJKL

14. 根据题目要求，设计出完整的程序。

(1) 在已给的 10 个数{0, 12, -9, 63, 54, 98, -65, 123, -12, 10}中，找出最大值及其下标并输出。

```

#include <iostream.h>
int k=0;
void input(int x[], int n)           //数组 x 用作形参,n 为数组元素的个数
{                                     //把输入的数据送到数组 x 中
    cout<<"输入"<<n<<"个整数: ";
    for(int i=0;i<n;i++) cin>>x[i];
}
int big(int b[],int n)               //数组 b 用作形参, n 为数组元素的个数
{                                     //从数组 b 中找出最大值, 并返回最大值
    int max=b[0];
    for(int i=1;i<n;i++)if (max<b[i]) { max=b[i]; k=i;}
    return max;
}
void main(void)
{
    int a[10],max;
    input(a,10);                     //调用函数输入 10 个数
    max=big(a,10);                   //数组名 a 用作实参, 调用函数求最大值
    cout<<max<<" "<<k<<endl;
}

```

(2) 在有序数列{-65, -12, -9, 0, 10, 12, 54, 63, 98, 123}中插入任意一个值 x (从键盘上读入), 使其仍然有序, 然后输出。

```

#include <iostream.h>
void main(void)
{
    int a[11]={-65,-12,-9,0,10,12,54,63,98,123};
    int x,k;
    for (int i=0;i<11;i++) cout<<a[i]<<' ';
    cout<<endl;
    cout<<"x=";
    cin>>x;
    i=0;
    while (a[i]<x && i<10) i++;
}

```

```

k=i;
cout<<k<<endl;
for (i=10;i>k;i--) a[i]=a[i-1];
a[k]=x;
for (i=0;i<11;i++) cout<<a[i]<<' ';
cout<<endl;
}
或者
#include <iostream.h>
void main(void)
{
int a[11]={-65,-12,-9,0,10,12,54,63,98,123};
int x,k;
for (int i=0;i<11;i++) cout<<a[i]<<' ';
cout<<endl;
cout<<"x=";
cin>>x;
a[10]=x;i=10
while (a[i-1]>a[i]&& i>0){
k=a[i-1];a[i-1]=a[i];a[i]=k;i--;
}
for (i=0;i<11;i++) cout<<a[i]<<' ';
cout<<endl;
}

```

(3) 不用 strcpy 函数，自己设计一个实现将一个字符数组中的字符串复制到另外一个数组中的函数，并对其正确性进行测试。

```

#include <iostream.h>
void strcpy(char a[],char b[])
{
int l=0;
while (a[l++]=b[l]);
}
void main(void)
{
char a[50],b[50]="abcde";
strcpy(a,b);
cout <<a<<"\n";
}

```

(4) 不用 strcat 函数，自己设计一个实现将一个字符数组中的字符串拼接到另一个字符数组中的函数，并对其正确性进行测试。

```

#include <iostream.h>
#include <string.h>
void strcat(char a[],char b[])
{

```

```

    int j,n;
    j=strlen(a);
    n=strlen(b);
    for (int k=0;k<=n;k++) a[j++]=b[k];
}
void main(void)
{
    char s1[50]="欲穷千里目, ";
    char s2[30]="更上一层楼。";
    strcat(s1,s2);
    cout <<s1<<"\n";
}

```

(5) 21 个人围成一个圈，编号依次为 1~21。从第 1 号开始报数，报到 5 的倍数的人离开，一直报下去直到最后只剩下 1 人，求出此人的编号。

```

#include <iostream.h>
void main(void)
{
    int p[21],n=0,flag,last;
    for (int i=0;i<21;i++){           // 数组元素赋初值
        p[i]=1;
        cout<<"* ";
    }
    cout<<endl;
    do {
        flag=0;
        for (i=0;i<21;i++){
            if (p[i]){
                flag=1;
                n++;
                if (n%5==0){
                    p[i]=0; last=i;
                    for (int j=0;j<21;j++) cout<< (p[j]?"* ":" ");
                    cout<<endl;
                }
            }
        }
    }
    while (flag);
    cout <<"最后一人的编号为: "<<last<<"\n";
}

```

(6) 输入 10 个字符串和一个待查找的字符串，然后从这 10 个字符串中查找并统计出与待查找字符串相同的字符串的个数。

用 strcmp 函数实现

(7) 输入一个 5×5 的数组，求出主、副对角线元素之和。

```

#include <iostream.h>

```



```

void main(void)
{
    int i,j;
    float a[5][5],sum=0;
    cout <<"输入一个 5×5 矩阵: ";
    for (i=0;i<5;i++)
        for (j=0;j<5;j++)
            cin >>a[i][j];
    for (i=0;i<5;i++)
        sum+=a[i][i]+a[i][4-i];
    cout <<"两对角线之和为: "<<sum<<"\n";
}

```

(8) 输入一个 5×6 的数组，求出每行中的最大值及其行、列下标。

```

#include <iostream.h>
void main(void)
{
    int i,j,k;
    float a[5][6],max;
    cout <<"输入一个 5×6 矩阵: ";
    for (i=0;i<5;i++)
        for (j=0;j<6;j++)
            cin >>a[i][j];
    for (i=0;i<5;i++){
        max=a[i][0]; k=0;
        for (j=0;j<6;j++)
            if (max<a[i][j]) { max=a[i][j]; k=j; }
        cout <<"第"<<i<<"行的最大值是: "<<max<<"\n";
    }
}

```

(9) 输入一行字符，统计出其中的单词个数，输入行中单词之间用一个或多个空格隔开。

```

#include <iostream.h>
#include <string.h>
void main()
{
    char s[200];
    int n = 1; i = 0;
    cout<<"输入一行字符串: ";
    cin.getline(s,100);
    for(i = 0; i <= strlen(s); i++)
    {
        if(s[i] != ' ' && s[i+1] == ' ')
            n++;
    }
    cout<<"单词个数为: "<<n<<endl;
}

```


7. 设有以下枚举类型说明语句:

```
enum weekday {Mon=1,Tues,Wed,Thurs,Fri,Sat,Sun=0}week;
```

对变量week不正确的赋值语句是__B__。

A) week= weekday(1) B) week=1 C) week=Mon D) week=(weekday)1

8. 设有以下说明语句:

```
struct test{
    int i; char ch; float f;
    union uu {char s[5]; int m[2];} ua;
} ex;
```

下列对成员m[1]的正确引用是__C__。

A) ex.m[1] B) ex.uu.m[1] C) ex.ua.m[1] D) ex.test.m[1]

9. 若有下面说明语句:

```
struct test{
    int i; char ch; float f;
    union un{char s[5]; int m[2];} u;
};
```

则sizeof(test)的值是__D__。

A) 12 B) 9 C) 22 D) 20

10. 有以下程序段:

```
struct abc{ int x, y;} h[2];
h[0].x=1; h[0].y=2; h[1].x=3; h[1].y=4;
```

则结构体成员h[0].x和h[1].y的值分别是__C__。

A) 1 2 B) 1 3 C) 1 4 D) 3 4

11. 以下程序的输出结果是__D__。

```
#include <iostream.h>
union myun{
    struct{
        int a,b,c;
    }s;
    int m;
} u;
void main(void)
{
    u.s.a=1; u.s.b=3; u.s.c=5;
    u.m=9;
    cout<<u.s.a<<endl;
}
```

A) 1 B) 3 C) 5 D) 9

12. 以下程序的输出结果是__C__。

```
#include <iostream.h>
void main(void)
{
    union baby {
        char name[10];
```

```

        int number;
    }b={"YangYang"};
    cout<<b.name<<" ";
    b.number=65;
    cout<<b.name<<" "<<b.number<<endl;
}

```

- A) YangYang YangYang 65 B) YangYang 65 65
 C) YangYang A 65 D) YangYang 65

13. 下列程序的输出结果是 B。

```

#include <iostream.h>
void main(void)
{
    enum tag{Up=1,Down,Left,Right}x=Up,y;
    enum tag z=Left; y=Down;
    cout<<x<<" "<<y<<" "<<z<<endl;
}

```

- A) Up Down Left B) 1 2 3 C) 0 1 2 D) Up Left Down

14. 编写完整的程序。

(1) 打印一个班组的通讯录。该通讯录包括：班级、学号、姓名、住址、电话、E-mail 等。

```

struct Address_List
{
    char clas[20];
    char student_id[40];
    char student_name[20];
    char student_address[100];
    char student_phone[20];
    char student_email[100];
};

```

(2) 输入 n 个人的编号、姓名、身高，然后按身高从小到大的顺序排列输出。

(3) 从键盘输入某人的姓名，然后从已知的通讯录中查找该人是否存在，并给出相应的信息。

```

strcmp(getchar, clas[i].student_name);

```

(4) 定义关于 12 个月的枚举类型，从键盘上输入 1~12 后输出对应的英文名称。

```

enum month {January =1, February, March, April, May, June, July, August, September,
October, November, December};
int month_number;
cout<<(month)month_number;

```

第八章

1. 在 int k=8, *p=&k 中, *p 的值是 C。
 A) 指针变量 p 的地址值 B) 变量 k 的地址值 C) 8 D) 无意义
2. 对于说明语句: int *p[10]; 正确的描述是 B。
 A) p 是指向数组中第 10 个元素的指针
 B) p 是具有 10 个元素的指针数组, 每个元素是一个 int 型指针

- C) p 是指向数组的指针
D) p[10]表示数组的第 10 个元素的值
3. 对于指针的运算, 错误的描述是 C。
- A) 可以用一个空指针赋值给某个指针变量
B) 一个指针可以加上一个整数
C) 两个指针可以进行加法运算
D) 两个指针在一定条件下, 可以进行相等或不相等的运算
4. 下列定义中, A 是定义指向数组的行指针。
- A) int (*p)[5] B) int *p[] C) int *p[5] D) (int *)p[5]
5. 设有说明语句: int *fun(); fun表示 B。
- A) 一个用于指向函数的指针变量 B) 一个返回值为指针型的函数名
C) 一个用于指向一维数组的行指针 D) 一个用于指向 int 型数据的指针变量
6. 设有说明语句: const char *ps; ps表示 C。
- A) 指向字符串的指针 B) 指向字符串的 const 型指针
C) 指向 const 型字符串的指针 D) 指向 const 型字符串的 const 型指针
7. 指针可以表示数组运算, 对于数组int a[5][5], 错误的表示是 B。
- A) *(a+2)[2] B) *(&[2][2]+2) C) (*(a+2)) D) *(*a+2)
8. 下列表示main函数的命令行参数中, 错误的表示形式是 C。
- A) main(int argv,char *argv[]) B) main(int arc,char **arv)
C) main(int argc,char *argv) D) main(int a,char *c[])
9. 设有说明语句: float x=6.2; 下列说明x的别名方法中, 正确的是 B。
- A) float &y=&x; B) float &y=x; C) float y=&x; D) float &y;
10. 设有说明语句: int m, n=2, *p=&n; 则能完成m=n赋值功能的语句是 A。
- A) m=*p; B) *p=*&n; C) m=&n; D) m=**p;
11. 下列引用调用中, A 是正确的调用方式。
- A) 形参是引用, 实参是变量 B) 形参是数组名, 实参是数组名
C) 形参是变量, 实参也是变量 D) 形参是指针, 实参是地址
12. 下列关于new运算符的描述中, B 是错误的。
- A) 创建基本类型对象时, 可以指定初值 B) 创建多个对象时, 可以指定初值
C) 创建的对象可以用 delete 删除 D) 可以动态创建单个对象或多个对象
13. 下列关于delete运算符的描述中, B 是错误的。
- A) 它必须作用于 new 返回的指针
B) 对一个 new 返回的指针可以使用多次 delete 运算符
C) 它可以作用于空指针
D) 指针名前可以用方括号 [], 表示释放指向多个对象首地址的指针
14. 填空题。
- (1)有说明语句: int *p; 则*p++运算首先访问 *p, 然后使 p 的值加 1。
(2)有说明语句: int *p; 则(*p)++运算首先访问 *p, 然后使 *p 的值加 1。
(3)把int型变量y定义为int型变量x的引用, 所使用的语句是 int &y = x;。
(4)若y是x的引用, 则对y的操作就是对 x 的操作。
(5)若y是x的引用, 则&y和&x的值是 相同, 即为变量 x 的地址。
(6)执行int *p=new int操作, 可以得到一个动态分配整型对象的 地址。
(7)执行int *p=new int[5]操作, 使p指向动态分配的整型数组中下标为 0 的元素。
(8)执行 delete p 操作将释放由p指向的动态分配对象的内存空间。

(9)有以下程序段:

```
int i;int &j=i;
i=8;j=i+1;
```

则i= 9 , j= 9 。

(10)若执行c:\command aaa.txt to bbb.cpp命令行, 对于int main(int argc,char *argv[]){ ...}, argc的值是 4 , argv[0] 的值是 c:\command argv[1] 的值是 aaa.txt , argv[argc-1] 的值是 bbb.cpp 。

15. 选择下列程序的输出结果。

(1)程序的输出结果是 C 。

```
#include <iostream.h>
void main(void)
{
    char *p="abcdefgh",*r;
    int *q;
    q=(int *)p; q++; r=(char*)q;
    cout<<r<<endl;
}
```

A) abcd B) a C) efgh D) b

(2)程序的输出结果是 B 。

```
#include <iostream.h>
void fun(int *a, int *b)
{ int *k; k=a; a=b; b=k; }
void main(void)
{
    int a=3, b=6, *x=&a, *y=&b;
    fun(x,y);cout<<a<<","<<b<<endl;
}
```

A) 3,3 B) 3,6 C) 6,3 D) 6,6

(3)程序的输出结果是 B 。

```
#include <iostream.h>
void amovep(int *p, int a[],int n)
{
    int i;
    for(i=0;i<n;i++){
        *p=a[i]; p++;
    }
}
void main(void)
{
    int *p,a[9]={1,2,3,4,5,6,7,8,9};
    p=new int[10]; amovep(p,a,9);
    cout<<p[2]<<","<<p[5] <<endl;
    delete []p;
}
```

A) 3,3 B) 3,6 C) 6,3 D) 6,6

(4)程序的输出结果是 D 。

```
#include <iostream.h>
sub(int *a,int n,int k)
{
    if(k<=n) sub(a,n/2,2*k);
    *a+=k;
}
void main(void)
{
    int x=0;
    sub(&x,8,1);
    cout<<x<<endl;
}
```

A) 8 B) 2 C) 6 D) 7

(5)程序的输出结果是 D 。

```
#include <iostream.h>
const int n=5;
void main(void)
{
    int a[n]={3,10,5,6,12};
    int *p1=a,*p2=a+n-1;
    while(p1<p2){
        int x=*p1;*p1=*p2;*p2=x;
        p1++;p2--;
    }
    for(int k=0;k<n;k++)cout<<*(a+k)<<" ";
    cout<<endl;
}
```

A) 3 6 5 10 12 B) 3 5 6 10 12
C) 12 10 6 5 3 D) 12 6 5 10 3

16. 编写完整的程序。

(1)设计一个函数 change，其功能是当调用 change(a,b,c)时，使得 b=a,c=b,a=c，要求用两种方法实现：

①传地址方式，②引用方式。

5. 设计一个函数：**exchange(float x,float y,float z)**，当调用 **exchange(a,b,c)**时，将 **a** 的内容设置给 **b**，**b** 的内容设置 **c**，**c** 的内容设置给 **a**。要求使用两种方法实现。

解：方法一：采用传地址方式，其代码如下：

```
void exchange(float *x,float *y,float *z)
{
    float *temp1,*temp2;
```

```

    temp1=*y;temp2=*z;
    *y=*x;*z=temp1;*x=temp2;
}

```

方法二：采用引用调用方式，其代码如下：

```

void exchange(float &x,float &y,float &z)
{
    float temp1,temp2;
    temp1=y;temp2=z;
    y=x;z=temp1;x=temp2;
}

```

或下面完整的程序：

```

#include <iostream.h>
void changel(int *p1,int *p2,int *p3)
{
    int temp;
    temp=*p1;
    *p1=*p2;
    *p2=*p3;
    *p3=temp;
}
void change2(int &p1,int &p2,int&p3)
{
    int temp;
    temp=p1;
    p1=p2;
    p2=p3;
    p3=temp;
}
void main(void)
{
    int a,b,c;
    cout <<"Input a,b and c: ";
    cin >>a>>b>>c;
    cout <<"a= "<<a<<", b= "<<b<<", c= "<<c<<"\n";
    changel(&a,&b,&c);
    cout <<"a= "<<a<<", b= "<<b<<", c= "<<c<<"\n";
    change2(a,b,c);
    cout <<"a= "<<a<<", b= "<<b<<", c= "<<c<<"\n";
}

```

(2)输入任意一个字符串，将其中的大写字母转换成小写字母。

ASCII 码值加 32

(3)已知全班 30 个学生的 6 门功课成绩，求全班 6 门功课的平均成绩，并按降序输出，要求用两种方

法实现：①传地址方式②引用方式。

ppt

(4)设计一个函数，求字符串的长度（指向字符串的指针作为函数的参数）。在主函数中输入一个字符串，并输出这个字符串及其长度。

【2.80】 参考答案：

```
#include <iostream.h>
int Strlen(char *s)
{
    char *p = s;
    while (*p)
        p++;
    return (p - s);
}
void main()
{
    char ch[100];
    cout << "Please input a string: ";
    cin >> ch;
    cout << "The result is " << Strlen(ch) << ".";
}
```

(5)定义一个指向字符串的指针数组，用一个函数完成 n 个不等长字符串的输入，用 new 运算符根据实际输入的字符串长度分配存储空间，依次使指针数组中的元素指向每一个输入的字符串。设计一个完成 n 个字符串排序的函数（在排序的过程中，要求只交换指向字符串的指针值，不交换字符串）。在主函数中完成将排序后的字符串输出。

```
#include <iostream.h>
#include <string.h>
void input(char *ps[ ],int n)
{
    char s[80];
    cout <<"请输入"<<n<<"个字符串：\n";
    for (int j=0;j<n;j++){
        cin.getline(s,80);
        ps[j]=new char[strlen(s)+1];
        strcpy(ps[j],s);
    }
}
```

```

void sort(char *ps[ ], int n)
{
    int j, k, l;
    char *t;
    for (j=0; j<n-1; j++) {
        l=j;
        for (k=j+1; k<n; k++)
            if (strcmp(ps[l], ps[k])>0) l=k;
        if (l!=j) {
            t=ps[j]; ps[j]=ps[l]; ps[l]=t;
        }
    }
}

void output(char *ps[ ], int n)
{
    cout <<"排序后的字符串: \n";
    for (int j=0; j<n; j++) {
        cout <<ps[j]<<" \n";
    }
}

void release(char *ps[ ], int n)
{
    cout <<"释放动态分配的存储空间! \n";
    for (int j=0; j<n; j++) {
        delete ps[j];
    }
}

void main(void)
{
    char *ps[20], s[80];
    int n;
    cout <<"请输入字符串个数(n<=20): ";
    cin >>n;
    cin.getline(s, 80);
    input(ps, n);
    sort(ps, n);
    output(ps, n);
    release(ps, n);
}

```

(6)输入一个字符串，串内有数字和非数字字符，如：

```
abc2345 345rrf678 jfkld945
```

将其中连续的数字作为一个整数，依次存放到另一个整型数组 **b** 中。如对于以上的输入，将 2345 存放到 **b[0]**，345 放入 **b[1]**，…。统计出字符串中的整数个数，并输出这些整数。要求在主函数中完成输入和输出，设计一个函数，把指向字符串的指针和指向整数的指针作为函数的参数，并完成从字符串中依次提取出整

数的工作。

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

int calculate(char array_char[ ], double array_double[ ])
{
    int length = strlen(array_char);
    int flag = 0;
    double num = 0;
    int j = 0;
    for(int i = 0; i < length; i++)
    {
        if(array_char[i] >= 48 && array_char[i] <= 57)
        {
            num = num * 10 + (array_char[i] - 48);
            flag = 1;
        }
        else if(flag == 1)
        {
            array_double[j++] = num;
            num = 0;
            flag = 0;
        }
    }
    if(flag == 1)
        array_double[j++] = num;
    return j;
}

void main()
{
    char array[100];
    double array_double[100];
    int length;
    cout << "请输入一串字符串，可含有数字和非数字字符： ";
    cin.getline(array, 100);
    length = calculate(array, array_double);
    for(int i = 0; i < length; i++)
        cout << array_double[i] << endl;
}
```

(7)建立一条无序链表，每一个结点包含：学号、姓名、年龄、C++成绩。由一个函数完成建立链表的工作，另一个函数完成输出链表上各结点值，一个函数完成释放链表结点占用的动态存储空间。

```
#include <iostream.h>
#include <iomanip.h>
struct node{
    char name[10];
```

```

    int id, age;
    float cpp;
    node *next;
};
node *create(void)
{
    node *p1,*p2,*head=0;
    int no;
    cout <<"产生一条无序链表, 请输入学号 (以 0 结束): ";
    cin >>no;
    while (no!=0) {
        p1=new node;
        p1->id=no;
        cout <<"请依次输入姓名、年龄、C++成绩: \n";
        cin >>p1->name>>p1->age>>p1->cpp;
        if (head==0) {
            head=p1; p2=p1;
        }
        else {
            p2->next=p1; p2=p1;
        }
        cout <<"请输入学号 (以 0 结束): ";
        cin >>no;
    }
    if (head) p2->next=0;
    return (head);
}
void output(node *h)
{
    node *p;
    p=h;
    cout <<"学号 姓名 C++成绩\n";
    while (p!=0) {
        cout <<setw(4)<<p->id<<setw(6)<<p->name<<setw(7)<<p->cpp;
        cout <<"\n";
        p=p->next;
    }
}
void deletechain(node *h)
{
    node *p1;
    while (h) {
        p1=h;
        h=h->next;
    }
}

```

```

        delete p1;
    }
}
void main(void)
{
    node *head;
    head=create( );
    output(head);
    deletechain(head);
}

```

下面的代码可参考，但要求略有不同。

【2.90】 参考答案：

```

#include <iostream.h>
struct student
{
    int age;
    float score;
    student* next;
};
int n; //动态分配结构 student 的个数
//-----新建链表-----
struct student *Create() //此函数返回一个指向链表头的指针
{
    struct student *head, *p1, *p2;
    n = 0;
    p1 = p2 =new student;
    cout << "Please input a student's age and score: ";
    cin>>p1->age>>p1->score;
    head = NULL;
    //当输入学生年龄为 0 时链表建立结束
    while (p1->age != 0)
    {
        n = n+1;
        if (n == 1)
            head = p1;
        else
            p2->next = p1;
        p2 = p1;
        p1 =new student;
    }
}

```

```

        cout << "Please input a student's age and score: ";
        cin >> p1->age >> p1->score;
    }
    p2->next = NULL;
    return head;
}
//-----输出链表-----
void print(struct student *head)
{
    struct student *p;
    p = head;
    while (p != NULL)
    {
        cout <<p->age << "    " << p->score;
        p = p->next;
    }
}
//-----删除链表中的某一结点-----
//-----根据年龄删除-----
struct student *Delete(struct student *head, int age)
{
    struct student *p1 ,*p2;
    if (head == NULL)
    {
        cout << "list null!";
        return(head);
    }
    p1 = head;
    while (age != p1->age && p1->next != NULL)
    {
        p2 = p1;
        p1 = p1->next;
    }
    //后移一个结点
    if (age == p1->age)
    {
        if (p1 == head)
            head = p1->next;
        else
            p2->next = p1->next;
    }
}

```

```

        delete p1;                                //释放节点 p1
        cout << "delete: " << age << endl;
    }
    else
        cout << age << "has not been found!" << endl;    //找不到该节点
    return(head);
}
//-----在链表中插入节点-----
struct student *insert(struct student *head, struct student *stud)
{
    struct student *p0, *p1, *p2;
    p1 = head;
    p0 = stud;
    if (head == NULL)
    {
        head = p0;
        p0->next = NULL;
    }
    else
        while ((p0->age > p1->age) && (p1->next != NULL))
        {
            p2 = p1;
            p1 = p1->next;
        }
    if (p0->age <= p1->age)
    {
        if (head == p1)
        {
            head = p0;
            p0->next = p1;
        }
        else
        {
            p2->next = p0;
            p0->next = p1;
        }
    }
    else
    {

```

```

        p1->next = p0;
        p0->next = NULL;
    }
    n=n+1; //插入节点, 节点数增加 1
    return head;
}
void main()
{
    student* s, *t;
    s = Create();
    print(s);
    s = Delete(s, 18);
    print(s);
    cout << "Please input a student's age and score: ";
    t = new student;
    cin >> t->age >> t->score;
    s = insert(s, t);
    print(s);
    delete t;
}

```

(8)建立一条有序链表，其它的要求与上一题相同，按 C++成绩的升序排序。
在插入一个节点时，要找到正确的插入位置。

第九章

1. 类的一般形式是什么？
2. 解释 `public` 和 `private` 的作用。类和公有成员与私有成员有什么区别？
3. 数据成员可以是公有的吗？成员函数可以是私有的吗？
4. 类与对象之间的关系如何？
5. 什么是 `this` 指针？它的主要作用是什么？
6. 阅读下面的类定义，找出程序中的错误，并说明错误原因。

```

class my{
public:
    my a; //类体中不能定义自身的对象作为数据成员
private:
    int x;
};

```

7. 定义一个描述猫的种类 `Cat`，包含 `age`、`weight`、`colour` 等属性以及对这些属性操作的方法。实现并测试这个类。

```

class Cat
{
    unsigned int age;
    double weight;

```



```

char color[20];
public:
    Cat(unsigned int age1, double weight1, char color1[])
    {
        .....
    }
};

```

8. 设计一个矩形类 CRectangle，要求有下述成员函数：

- 1) Move: 从一个位置移动到另一个位置。
- 2) Size: 改变矩形的大小。
- 3) Where: 返回矩形左上角的坐标值。
- 4) Area: 计算矩形面积。

【3.51】 参考答案：

```

class CRectangle
{
public:
    void Move(int x, int y);
    void Size(int widthNum, int highNum);
    void Where(int &x, int &y);
    double Area();
private:
    int position_x, position_y, width, high;
};
void CRectangle::Move(int x, int y)
{
    position_x = x;

    position_y = y;
}
void CRectangle::Size(int widthNum, int highNum)
{
    width = widthNum;
    high = highNum;
}
void CRectangle::Where(int &x, int &y)
{
    x = position_x;
    y = position_y;
}
double CRectangle::Area()
{
    return width * high;
}

```

9. 基于计算器的使用经验，定义类 Calculator 实现计算器的常用功能，例如，加、减、乘、除、平方根、常用对数、自然对数、幂、正弦、余弦等（提示：可以利用 C++提供的库函数来求解常用对数、自然对数等运算）。

【3.53】 参考答案：

```
#include <math.h>
#include <stdlib.h>
class Calculator
{
public:
    double Add(double x, double y);           //求 x+y
    double Sub(double x, double y);          //求 x-y
    double Mul(double x, double y);          //求 x*y
    double Div(double x, double y);          //求 x/y
    double Sin(double x);                     //求 sin(x)
    double Cos(double x);                     //求 cos(x)
    double Tan(double x);                     //求 tan(x)
    double Exp(double x);                     //求 e 的 x 次方
    double Fabs(double x);                    //求 x 的绝对值
    double LogE(double x);                    //求以 e 为底的对数
    double Log10(double x);                  //求以 10 为底的对数
    double Pow(double x, double y);          //求 x 的 y 次方
```

```

    double Sqrt(double x);           //求 x 的平方根
};
double Calculator::Add(double x, double y)
{
    return x + y;
}
double Calculator::Sub(double x, double y)
{
    return x - y;
}
double Calculator::Mul(double x, double y)
{
    return x * y;
}
double Calculator::Div(double x, double y)
{
    if (y == 0) exit(0);           //①
    return x / y;
}
double Calculator::Sin(double x)
{
    return sin(x);
}
double Calculator::Cos(double x)
{
    return cos(x);
}
double Calculator::Tan(double x)
{
    return tan(x);
}
double Calculator::Exp(double x)
{
    return exp(x);
}
double Calculator::Fabs(double x)
{
    return fabs(x);
}
}

```

```

double Calculator::LogE(double x)
{
    return log(x);
}
double Calculator::Log10(double x)
{
    return log10(x);
}
double Calculator::Pow(double x, double y)
{
    return pow(x, y);
}
double Calculator::Sqrt(double x)
{
    if (x < 0)    return 0;
    return sqrt(x);
}

```

注释：当除数为 0 时，由语句①退出程序。

10. 在 C++ 类库中，对不同类型的数求其绝对值需要使用不同的函数。求整数的绝对值使用函数 `abs`，求长整型的绝对值使用函数 `labs`，求浮点数的绝对值使用函数 `fabs`。要求定义一个类 `ABS`，并重载成员函数 `abs`，使 `abs` 完成对不同类型数据求绝对值的运算。

【3.54】 参考答案：

```

class ABS
{
public:
    int abs(int a)
    {
        return a > 0 ? a : -a;
    }
    long abs(long a)
    {
        return long(a > 0 ? a : -a);
    }
    double abs(double a)
    {
        return a > 0 ? a : -a;
    }
};

```

第十章

1. 举例说明构造函数和析构函数有什么作用。
2. 什么叫做拷贝构造函数？拷贝构造函数何时被调用？
3. 拷贝构造函数与赋值运算符(=)有何不同？
4. 指出以下程序段中的错误，并说明错误原因。

```
class Time{
public:
    Time();
    Time(int h=0,int m=0,int s=0);
    void ~Time(int);
private:
    int hours,minutes,seconds;
};
```

- 1) 缺省的构造函数不唯一
 - 2) 析构函数不可说明返回值类型及参数
5. 指出以下程序中的错误，说明错误原因并改正。

```
#include <iostream.h>
class Test{
public:
    Test(int i) { ptr=new int(i); }
    ~Test(){ delete ptr; }
private:
    int *ptr;
};
void main(void)
{
    Test obj1(1);
    Test obj2(obj1);
} // 运行时错误，重复释放内存
```

需定义拷贝的构造函数

6. 写出以下程序的运行结果。

```
#include <iostream.h>
class ClassOne{
    int i;
public:
    ClassOne();
    ClassOne(int value);
    void Print();
    ~ClassOne();
};
class ClassTwo {
```

```

        int i;
public:
    ClassOne myclass;
    ClassTwo();
    ClassTwo(int value);
    void Print();
    ~ClassTwo();
};
ClassOne::ClassOne()
{
    cout << "This is constructor of ClassOne!" << endl;
    i = 0;
}
ClassOne::ClassOne(int value)
{
    cout << "This is constructor of ClassOne!" << endl;
    i = value;
}
void ClassOne::Print()
{    cout << "The i of ClassOne is " << i << endl; }
ClassOne::~ClassOne()
{    cout << "This is destructor of ClassOne!" << endl; }
ClassTwo::ClassTwo()
{
    cout << "This is constructor of ClassTwo!" << endl;
    i = 0;
}
ClassTwo::ClassTwo(int value)
{
    cout << "This is constructor of ClassTwo!" << endl;
    i = value;
}
ClassTwo::~ClassTwo()
{    cout << "This is destructor of ClassTwo!" << endl;}
void ClassTwo::Print()
{    cout << "The i of ClassTwo is " << i << endl; }
void main(void)
{
    ClassTwo myclass(10);
    myclass.Print();
}

```

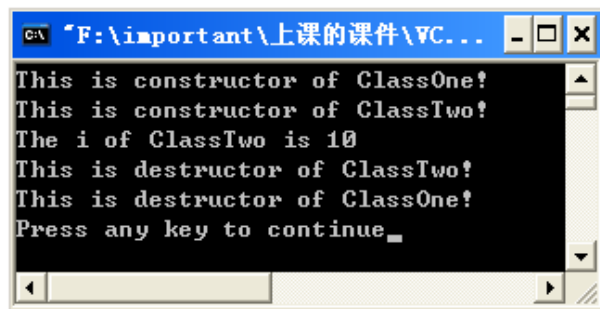
```

This is constructor of ClassOne!
This is constructor of ClassTwo!
The i of ClassTwo is 10

```

This is destructor of ClassTwo!

This is destructor of ClassOne!



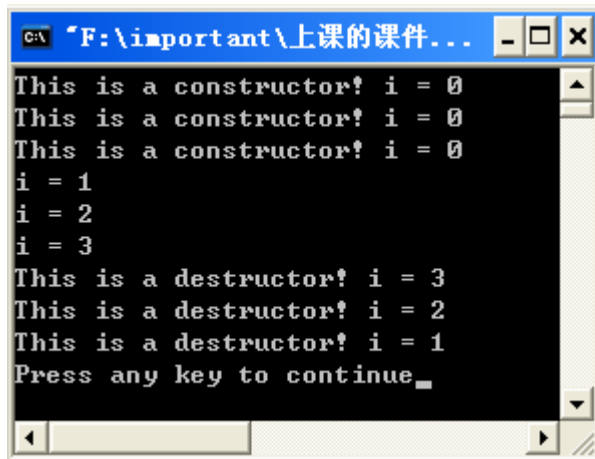
```
C:\ "F:\important\上课的课件\VC..."
This is constructor of ClassOne!
This is constructor of ClassTwo!
The i of ClassTwo is 10
This is destructor of ClassTwo!
This is destructor of ClassOne!
Press any key to continue_
```

7. 写出以下程序的运行结果。

```
#include <iostream.h>
class MyClass{
    int i;
public:
    void SetValue(int val);
    MyClass();
    ~MyClass();
};
MyClass::MyClass()
{
    i = 0;
    cout << "This is a constructor! i = " << i << endl;
}
void MyClass::SetValue(int val)
{
    i = val;
    cout << "i = " << i << endl;
}
MyClass::~MyClass()
{    cout << "This is a destructor! i = " << i << endl; }
void main(void)
{
    int k;
    MyClass my[3], *p;
    p = my;
    for (k = 0; k < 3; k++){
        p->SetValue(k + 1);
        p++;
    }
}
```

This is a constructor! i=0

```
This is a constructor! i=0
This is a constructor! i=0
i=1
i=2
i=3
This is a destructor! i=3
This is a destructor! i=2
This is a destructor! i=1
```



```
C:\ "F:\important\上课的课件..."
This is a constructor! i = 0
This is a constructor! i = 0
This is a constructor! i = 0
i = 1
i = 2
i = 3
This is a destructor! i = 3
This is a destructor! i = 2
This is a destructor! i = 1
Press any key to continue.
```

8. 写出以下程序的运行结果。

```
#include <iostream.h>
class MyClass{
    int i;
public:
    void SetValue(int val);
    MyClass();
    ~MyClass();
};
MyClass::MyClass()
{
    i = 0;
    cout << "This is a constructor! i = " << i << endl;
}
void MyClass::SetValue(int val)
{
    i = val;
    cout << "i = " << i << endl;
}
MyClass::~MyClass()
{    cout << "This is a destructor! i = " << i << endl; }
void main(void)
{
```



```
MyClass *my1[3];  
int k;  
for (k = 0; k < 3; k++) my1[k] = new MyClass;  
for (k = 0; k < 3; k++) delete my1[k];  
}
```

This is a constructor! i=0
This is a constructor! i=0
This is a constructor! i=0
This is a destructor! i=0
This is a destructor! i=0
This is a destructor! i=0



```
C:\ F:\important\上课的课件...  
This is a constructor! i = 0  
This is a constructor! i = 0  
This is a constructor! i = 0  
This is a destructor! i = 0  
This is a destructor! i = 0  
This is a destructor! i = 0  
Press any key to continue
```

9. 试写出表示时间的类 `Time` 的定义及其成员函数的实现。要求：在 `Time` 类中包含 3 个数据成员，即时(hour)、分(minute)和秒(second)。在类的成员函数中应包括存（用 `set` 开头）、取（用 `get` 开头）各个数据成员和完整时间的操作；按上下午各 12 小时或按 24 小时打印时间的函数；有缺省值的构造函数等。缺省值以 0 时 0 分 0 秒为准。

10.3 class Time

```
public:
    Time (int h=0, int m=0, int s=0)
        : hr (h), min (m), sec (s) { normalize (); }
    int hours () { return hr; }
    int minutes () { return min; }
    int seconds () { return sec; }
    void advance (int h=0, int m=0, int s=1);
    void reset (int h=0, int m=0, int s=0);
    void print () { cout << hr << ":" << min << ":" << sec; }
private:
    int hr, min, sec;
    void normalize ();
};

void Time::normalize ()
{ min += sec/60;
  hr += min/60;
  hr %= 24;
  min %= 60;
  sec %= 60;
}

void Time::advance (int h, int m, int s)
{ hr += h;
  min += m;
  sec += s;
  normalize ();
}

void Time::reset (int h, int m, int s)
{ hr = h;
  min = m;
  sec = s;
  normalize ();
}
```

10. 定义整数集合类 IntSet。该类包括以下成员函数：

intSet();	//类的构造函数，根据需要可以定义多个构造函数
Empty();	//清空该整数集合
IsEmpty();	//判断整数集合是否为空
IsMemberOf();	//判断某个整数是否在该整数集合内
Add();	//增加一个整数到整数集合中
Sub();	//从整数集合中删除一个整数元素
IsEqual();	//判断两个集合是否相等
Intersection();	//求两个整数集合的交集
Merge();	//求两个整数集合的并集
Print();	//依次打印该整数集合

该类还包括以下成员变量：

int element[100];	//保存整数集合中的数据
int EndPosition;	//指示整数集合的最后一个元素位置

注意：整数集合中不允许有相同元素存在。另外，对于函数的参数及其返回类型，根据需要自己定义。

【4.54】 参考答案:

```
#include <iostream.h>
class intSet
{
public:
    intSet();
    intSet(int a[], int size);
    void Empty();
    bool IsEmpty();
    bool IsMemberOf(int a);
    bool Add(int a);
    bool Sub(int a);
    bool IsEqual(intSet& set);
    intSet Intersection(intSet& set);
    intSet Merge(intSet& set);
    void Copy(intSet& set);
    void Print();
private:
    int element[100];
    int EndPosition;
};
intSet::intSet()
{
    //初始化整数集合元素
    for (int i = 0; i < 100; i++)
```

```

        element[i] = 0;
        EndPosition = -1;
    }
intSet::intSet(int a[], int size )
{
    if (size >= 100)
        EndPosition = 99;
    else
        EndPosition = size - 1;
    for (int i = 0; i <= EndPosition; i++)
        element[i] = a[i];
    for (i = size; i < 100; i++)
        element[i] = 0;
}
void intSet::Empty()
{
    //清空整数集合元素
    for (int i = 0; i < 100; i++)
        element[i] = 0;
    EndPosition = -1;
}
bool intSet::IsEmpty()
{
    //判断整数集合是否为空
    if (EndPosition == -1)
        return true;
    else
        return false;
}
bool intSet::IsMemberOf(int a)
{
    for (int i = 0; i <= EndPosition; i++)
        if (element[i] == a)
            return true;
    return false;
}
bool intSet::Add(int a)
{
    //能否增加一个元素，需要知道

```

```

//1.该元素是否存在; 2.整数集合是否已经存满
if (IsMemberOf(a))
    return true;
else
    if (EndPosition == 99)
        return false;
    else
    {
        EndPosition++;
        element[EndPosition] = a;
        return true;
    }
}
bool intSet::Sub(int a)
{
    //能否删除一个元素, 需要知道整数集合是否已存在该元素
    int pos;
    if (!IsMemberOf(a))
        return false;
    for (int i = 0; i <= EndPosition; i++)
        if (element[i] == a)
        {
            pos = i;
            break;
        }
    for (i = pos; i < EndPosition; i++)
        element[i] = element[i+1];
    EndPosition--;
    return true;
}
bool intSet::IsEqual(intSet& set)
{
    //判断两个整数集合是否相等
    bool Equal;
    if (EndPosition != set.EndPosition)
        return false;
    for (int i = 0; i <= EndPosition; i++)
    {
        Equal = false;

```

```

        for (int j = 0; j <= set.EndPosition; j++)
            if (element[i] == set.element[j])
            {
                Equal = true;
                break;
            }
        if (!Equal)
            return false;
    }
    return true;
}

intSet intSet::Intersection(intSet& set)
{
    //求两个整数集合的交集
    int a[100], size = 0;
    for (int i = 0; i <= EndPosition; i++)
        for (int j = 0; j <= set.EndPosition; j++)
            if (element[i] == set.element[j])
            {
                a[size] = element[i];
                size++;
                break;
            }
    return intSet(a, size);
}

intSet intSet::Merge(intSet& set)
{
    //求两个整数集合的并集
    int a[100], size;
    for (int i = 0; i <= EndPosition; i++)
        a[i] = element[i];
    size = EndPosition + 1;
    for (i = 0; i <= set.EndPosition; i++)
        if (IsMemberOf(set.element[j]))
            continue;
        else
        {
            if (size == 99)
                break;

```

```

        a[size] = set.element[i];
        size++;
    }
    return intSet(a, size);
}
void intSet::Copy(intSet& set)
{
    for (int i = 0; i <= set.EndPosition; i++)
        element[i] = set.element[i];
    EndPosition = set.EndPosition;
}
void intSet::Print()
{
    for (int i = 0; i <= EndPosition; i++)
        cout << "element[" << i << "] = " << element[i] << endl;
}

```

第十一章

1. 比较类的三种继承方式 `public`、`private` 和 `protected` 之间的差别。
2. 派生类构造函数和析构函数的执行顺序是如何规定的？
3. 多继承在什么情况下会产生二义性？如何解决二义性？
4. 在类的派生中为什么要引入虚基类？虚基类构造函数的调用实现是如何规定的？
5. 根据基类的定义和测试程序，完成下面派生类的定义。

//类定义部分

```

class Point{
    int X,Y;
public:
    Point(int i=0,int j=0) {X=i; Y=j;}
    void Move(int xOff,int yOff) {X+=xOff; Y+=yOff;}
    int GetX() {return X;}
    int GetY() {return Y;}
};

class Rectangle : private Point{
    int W,H;
public:
    Rectangle(int x,int y,int w,int h) ①:Point(x, y) { ②W=w;H=h; }
    void Move(int xOff,int yOff) { ③Point::Move(xOff, yOff); }
    int GetX() { ④return Point::GetX(); }
    int GetY() { ⑤return Point::GetY(); }
}

```

```

        int GetW() { ⑥return W;       }
        int GetH() { ⑦return H;       }
};
//测试程序
#include <iostream.h>
#include <math.h>
void main(void)
{
    Rectangle rect(1,2,3,4);
    rect.Move(5,6);
    cout<<"The data of rect(X,Y,W,H): ("<<rect.GetX()<<","<<rect.GetY()<<","
        <<rect.GetW()<<","<<rect.GetH()<<")"<<endl;
}

```

6. 在下列给定的继承结构中，判断语句①~⑤是否有二义性。

```

class Base1{
public:
    int x;
    int fun1();
    int fun2();
    int fun2(int);
    int fun3();
};
class Base2{
    int x;
    int fun1();
public:
    char fun2();
    int fun3();
};
class Derived:public Base1,public Base2{
};
void main(void)
{
    Derived *ptr;
    ptr->x=1;           //① 有
    ptr->fun1();        //② 有
    ptr->fun2();        //③ 有
    ptr->fun2(10);     //④ 有
    ptr->fun3();        //⑤ 有
}

```

7. 写出以下程序的运行结果。

```

#include <iostream.h>
class Data{
    int x;

```



```

public:
    Data(int x)
    {
        Data::x=x;
        cout<<"Data cons."<<endl;
    }
    ~Data() { cout<<"Data des."<<endl; }
};
class Base{
    Data d1;
public:
    Base(int x):d1(x) { cout<<"Base cons."<<endl; }
    ~Base() { cout<<"Base des."<<endl; }
};
class Derived:public Base{
    Data d2;
public:
    Derived(int x):Base(x),d2(x) { cout<<"Derived cons."<<endl; }
    ~Derived() { cout<<"Derived des."<<endl; }
};
void main(void)
{   Derived obj(5); }

```

Data cons.
Base cons.
Data cons.
Derived cons.
Derived des.
Data des.
Base des.
Data des.

```

C:\Program Files\Mi...
Data cons.
Base cons.
Data cons.
Derived cons.
Derived des.
Data des.
Base des.
Data des.
Press any key to continue

```

8. 写出以下程序的运行结果。

```

#include <iostream.h>
class instrument{

```

```

public:
    void display() const { cout<<"instrument::display"<<endl; }
};
class piano:public instrument{
public:
    void display() const { cout<<"piano::display"<<endl; }
};
class guitar:public instrument{
public:
    void display() const{ cout<<"guitar::display"<<endl; }
};
void tone(instrument & i)
{ i.display(); }
void main(void)
{
    guitar guitar1;
    tone(guitar1);
    piano piano1;
    tone(piano1);
}
instrument::display
instrument::display

```

```

C:\Program Files\...
instrument::display
instrument::display
Press any key to continue

```

9. 建立普通的基类 **Building**，用来存储一座楼房的层数、房间数以及它的总平方米数。建立派生类 **House**，继承 **Building**，并存储下面的内容：卧室与浴室的数量。另外，建立派生类 **Office**，继承 **Building**，并存储灭火器与电话的数目等。

【6.74】 参考答案:

```
#include <iostream.h>
class Building
{
public:
    Building() { floor=room=0; area=0; }
    Building(int f,int r,double a)
    {
        floor=f;
        room=r;
        area=a;
    }
    void Print() const
    {
        cout<<"The building's specification:"<<endl;
        cout<<"Floor: "<<floor<<"\tRoom: "<<room<<"\tTotal Area: "<<area<<endl;
    }
protected:
    int floor,room;
```

```

    double area;
};
class House : public Building
{
public:
    House() { bedroom=bathroom=0; }
    House(int f,int r,double a,int be,int ba) : Building(f,r,a)
    {
        bedroom=be;
        bathroom=ba;
    }
    void Print() const
    {
        Building::Print();
        cout<<"Including..."<<"\tBedroom: "<<bedroom
            <<"\tBathroom: "<<bathroom<<endl;
    }
private:
    int bedroom,bathroom;
};
class Office : public Building
{
public:
    Office() { fireExtinguisher=telephone=0; }
    Office(int f,int r,double a,int fe,int te) : Building(f,r,a)
    {
        fireExtinguisher=fe;
        telephone=te;
    }
    void Print() const
    {
        Building::Print();
        cout<<"Including..."<<"\tFire Extinguisher: "<<fireExtinguisher
            <<"\tTelephone: "<<telephone<<endl;
    }
private:
    int fireExtinguisher,telephone;
};
void main()

```

```
{  
    House h(3,10,312.45,6,3);  
    Office o(20,800,22000,200,1400);  
    h.Print();  
    o.Print();  
}
```

10. 大学有这样几类人员：学生、教师、职员和在职读书的教师。给出这几类人员的类描述。提示：先提取公共的信息构成基类，由基类派生出各类人员。

【6.75】 参考答案:

```
,
#include <iostream.h>
#include <string.h>
class Person
{
public:
    Person(const char* n)
    {
        name=new char[strlen(n)+1];
        strcpy(name,n);
    }
    void Print() const { cout<<"Name: "<<name<<endl; }
protected:
    char* name;
};
class Student : virtual public Person
{
public:
    Student(const char* n,const char * m) : Person(n)
    {
        major=new char[strlen(m)+1];
        strcpy(major,m);
    }
    void Print() const
    {
        Person::Print();
        cout<<"Major: "<<major<<endl;
    }
protected:
    char* major;
};
class Staff : virtual public Person
{
```

```

public:
    Staff(const char* n,const char* d) : Person(n)
    {
        dept=new char[strlen(d)+1];
        strcpy(dept,d);
    }
    void Print() const
    {
        Person::Print();
        cout<<"Department: "<<dept<<endl;
    }
protected:
    char* dept;
};
class Teacher : public Staff
{
public:
    Teacher(const char* n,const char* d,const char* l) : Person(n),Staff(n,d)
    {
        lesson=new char[strlen(l)+1];
        strcpy(lesson,l);
    }
    void Print() const
    {
        Staff::Print();
        cout<<"Lesson: "<<lesson<<endl;
    }
protected:
    char* lesson;
};
class StudentTeacher : public Student, public Teacher
{
public:
    StudentTeacher(const char* n,const char* m,const char* d,const char* l)
        : Person(n),Student(n,m),Teacher(n,d,l) {}
    void Print() const
    {
        Student::Print();
        cout<<"Department: "<<dept<<endl;
    }
};

```

```

        cout<<"Lesson: "<<lesson<<endl;
    }
};
void main()
{
    Student stu("Mike","Software Engineering");
    Staff sta("Jason","Management");
    Teacher t("Tim","Computer","C++");
    StudentTeacher st("Sam","Computer Application","Computer","C++");
    stu.Print();
    sta.Print();
    t.Print();
    st.Print();
}

```

注释：注意虚基类及其保护成员的使用。

第十二章

1. 分析使用友元函数的利弊，说明友元函数与成员函数在使用上的差别。
2. 说明友元函数和友元类的区别和作用？
3. 虚函数的主要用途是什么？
4. 静态的数据成员与一般的数据成员有什么区别？
5. 静态的成员函数与一般的成员函数有什么区别？如何调用静态成员函数？
6. 写出以下程序的执行结果：

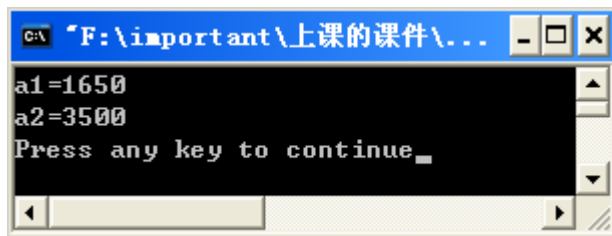
```

#include <iostream.h>
class A{
    int a,b;
    static int c;
public:
    A(int x) { a=x;}
    void f1(float x){b=a * x; }
    static void setc(int x) { c=x;}
    int f2() { return a+b+c;}
};
int A::c =100;
void main(void)
{
    A a1(1000),a2(2000);
    a1.f1(0.25); a2.f1(0.55);
    A::setc(400);
    cout<<"a1="<<a1.f2()<<"\n"<<"a2="<<a2.f2()<<"\n";
}

```


a1=1650

a2=3500



```
C:\> "F:\important\上课的课件\..."
a1=1650
a2=3500
Press any key to continue_
```

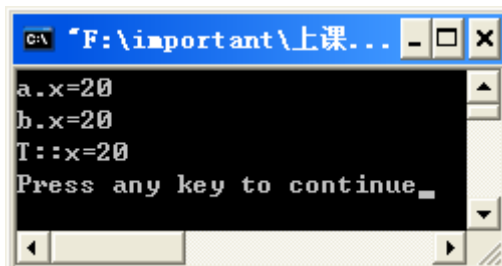
7. 写出以下程序的执行结果:

```
#include <iostream.h>
class T{ public: static int x; };
int T::x=1;
void main(void )
{
    T a,b; a.x=10; b.x=20;
    cout<<"a.x="<<a.x<<"\n"<<"b.x="<<b.x<<"\n"<<"T::x="<<T::x<<"\n";
}
```

a. x=20

b. x=20

T::x=20



```
C:\> "F:\important\上课..."
a.x=20
b.x=20
T::x=20
Press any key to continue_
```

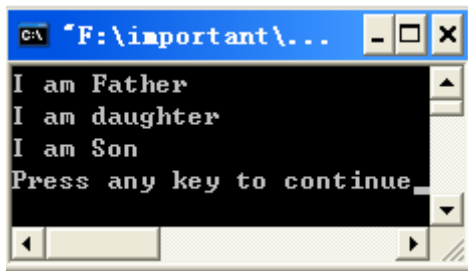
8. 写出以下程序的执行结果:

```
#include<iostream.h>
class Father{
public:
    virtual void answer() {cout<<"I am Father\n";}
};
class Daughter:public Father{
public:
    void answer(){cout<<"I am daughter\n";}
};
class Son:public Father{
public:
    void answer(){cout<<"I am Son\n";}
};
void main(void)
{
    Father father;
```

```

    Daughter daughter;
    Son son;
    Father *who;
    who=&father;who->answer();
    who=&daughter;  who->answer();
    who=&son;      who->answer();
}
I am Father
I am daughter
I am Son

```



9. 建立一条双向无序链表，结点数据包括：姓名，地址和工资。完成链表的基本操作。
10. 建立一条双向有序链表，结点数据包括：姓名，地址和工资。按工资从小到大的顺序排序。完成链表的基本操作。
11. 建立一条单向有序链表，结点为同学通讯录，结点数据包括：姓名，地址，邮编，电话，E-mail。完成通讯录的插入、查找、删除、列表输出等操作。
12. 指向类中数据成员的指针变量与一般的指针变量有什么不同？
13. 指向类中成员函数的指针变量与一般的指向函数的指针变量有什么不同？

第十三章

1. 什么是运算符重载？重载运算符的目的是什么？
2. 用成员函数实现运算符重载与用友元函数实现运算符重载，在定义和使用上有什么不同？
3. C++中的所有运算符是否都可以重载？凡是能用成员函数重载的运算符是否均能用友元函数重载？
4. 转换函数的作用是什么？如何使用转换函数？
5. 写出以下程序的执行结果。

```

#include <iostream.h>
class A{
    int x;
public:
    A(int a=0){ x=a;cout<<"x="<<x<<"\t"<<"构造函数\n"; }
    A(A & e){ x = e.x;cout<<"x="<<x<<"\t"<<"拷贝构造函数\n";}
    A & operator=(A & e)
    {
        x=e.x; cout<<"x="<<x<<"\t"<<"operator=\n";
        return *this;
    }
}

```

```

operator int()
{
    cout<<"x="<<x<<"\t"<<"operator int\n";
    return x;
}
};
void main(void)
{
    A x1(50);
    A x2 = 100;
    x1 = x2;
    A x3 = x1;
    int i = x1;
    x2 = (A)200;
}
x=50 构造函数
x=100 构造函数
x=100 operator=
x=100 拷贝构造函数
x=100 operator int
x=200 构造函数
x=200 operator=

```

```

C:\ F:\important\上课的...
x=50 构造函数
x=100 构造函数
x=100 operator=
x=100 拷贝构造函数
x=100 operator int
x=200 构造函数
x=200 operator=
Press any key to continue

```

6. 将以下程序输入计算机，执行该程序，分析程序输出的结果。

```

#include <iostream.h>
#include <string.h>
class String{
    char str[200];
public:
    String(char *p){strcpy(str,p);}
    String(){str[0]='\0';}
    String &operator =(String );
    friend String operator +(String &,String &);
    friend ostream &operator<<(ostream &,String &);
};

```

```

String &String::operator = (String p)
{
    strcpy(str,p.str); return *this; }
String operator + (String &st1,String &st2)
{
    String tmp;
    strcpy(tmp.str,st1.str);strcat(tmp.str,st2.str);
    return tmp;
}
ostream &operator<<(ostream &os,String &str)
{
    os<<str.str<<endl; return os; }
void main(void)
{
    String a("This "),b("C++ "), c("test.");d;
    d=a+b+c;
    cout<<d<<'\n';
}
This C++ test.

```



7. 定义一个复数类，通过重载运算符： $=$ 、 $+=$ 、 $-=$ 、 $+$ 、 $-$ 、 $*$ 、 $/$ 、 $==$ 、和 $!=$ ，直接实现两个复数对象之间的各种运算。编写一个完整的程序（包括测试以上各种运算符正确性的程序部分）。

8. 定义一个学生类，数据成员包括：姓名，学号，C++、数学、英语和物理的成绩。重载运算符“<<”和“>>”，实现学生类对象的直接输入和输出。增加转换函数，实现姓名和总成绩的转换。设计一个完整的程序，验证成员函数和重载运算符的正确性。

9. 定义描述平面上一个点的类 `point` 及其派生的直线类 `line`。在点类中重载运算符“++”和“--”，并区分这两种运算符的前置和后置运算。在直线类中重载运算符“==”和“!=”，分别判断两直线的长度是否相等或不等。设计一个完整的程序，能够测试重载运算符的正确性。

10. 定义一个包含实数数组及其元素个数的类，通过重载下标运算符来实现判下标是否出界的检查和赋值。

11. 通过重载函数调用运算符，分别实现二维和三维数组下标的合法性检查。二维数组为整数，三维数组为实数。

12. 完善字符串类，增加以下运算符的重载： $+=$ 、 $-=$ 、 $==$ 、 $!=$ 。在主函数中增加测试以上运算符正确性的程序部分。