习题 1 及其解答

1.1 选择题

1. 一个最简单的C++程序，可以只有一个(　c　)。

(a) 库函数　　(b) 自定义函数　　(c) main函数　　(d) 空函数

2. 用C++语言编制的源程序要变为目标程序必须要经过(　d　)。

(a) 解释　　(b) 汇编　　(c) 编辑　　(d) 编译

3. C++程序中的简单语句必须以(　b　)结束。

(a) 冒号　　(b) 分号　　(c) 空格　　(d)花括号

4. 有说明 int a=0; double x=5.16; 以下语句中，(　c　)属于编译错误。

(a) x=a/x;　　(b) x=x/a;　　(c) a=a%x;　　(d) x=x*a;

5. 执行C++程序时出现的“溢出”错误属于(　c　)错误。

(a) 编译　　(b) 连接　　(c) 运行　　(d) 逻辑

6. 下列选项中，全部都是C++关键字的选项为(　c　)。

(a) while IF Static　　(b) break char go

(c) sizeof case extern　　(d) switch float integer

7. 按C++标识符的语法规定，合法的标识符是(　a　)。

(a) _abc　　(b) new　　(c) π　　(d) “age“

8. C++语句中，两个标识符之间(　a　)不能作为C++的分隔符。

(a) 数字　　(b) ;　　(c) :　　(d) +

9. 下列正确的八进制整型常量表示是(　b　)。

(a) 0a0　　(b) 015　　(c) 080　　(d) 0x10

10. 下列错误的十六进制整型常量表示是(　c　)。

(a) 0x11　　(b) 0xaf　　(c) 0xg　　(d) 0x1f

11. 在下列选项中，全部都合法的浮点型数据的选项为(　b　)。

(a) -1e3.5　15.　2e-4　　(b) 12.34　-1e+5　0.1E-12

(c) 0.2e-2　-12345.　e-5　　(d) 5.0e(1+4)　0.1　8e+2

12. 下列正确的字符常量为(　d　)。

(a) “a”　　(b) ‘name’　　(c) a　　(d) ‘\101’

13. 下列选项中，(　d　)不能交换变量a和b的值。

(a) t=b;　b=a;　a=t;　　(b) a=a+b;　b=a-b;　a=a-b;

(c) t=a;　a=b;　b=t;　　(d) a=b;　b=a;

14. 关于下列语句叙述错误的是(　a　)。

```
int i=10,  *p=&i;
```

(a) p的值为10　　(b) p指向整型变量i

(c) *p表示变量i的值　　(d) p的值是变量i的地址

15. 有以下变量说明，下面不正确的赋值语句是(　b　)。

```
int a=5, b=10, c; int *p1 = &a, *p2 = &b;
```

(a) *p2 = b ;　　(b) p1 = a ;

(c) p2 = p1 ;　　(d) c = *p1 *(*p2) ;

16. 有以下变量说明，下面正确的语句是(　b　)。

```
int a=10, b;  int &pa=a, &pb=b;
```

(a) &pb = a;　　(b) pb = pa;　　(c) pb = &pa;　　(d) *pb = *pa;

17. 执行下面语句序列后，a和b的值分别为(　b　)。

```
int a = 5 , b = 3 , t ;
int &ra = a ;
int &rb = b ;
t = ra ; ra = rb ; rb = t ;
```

(a) 3和3　　(b) 3和5　　(c) 5和3　　(d) 5和5

18. 在下列运算符中，(d)优先级最高。

(a) <=　　(b) *=　　(c) +　　(d) *

19. 在下列运算符中，(d)优先级最低。

(a) !　　(b) &&　　(c) !=　　(d) ? :

20. 设 int i=1, j=2; 则表达式 i+++j 的值为(c)。

(a) 1　　(b) 2　　(c) 3　　(d) 4

21. 设 int i=1, j=2; 则表达式 ++i+j 的值为(d)。

(a) 1　　(b) 2　　(c) 3　　(d) 4

22. 在下列表达式选项中，(c)是正确。

(a) ++(a++)　　(b) a++b　　(c) a+++b　　(d) a++++b

23. 已知 int i=0, j=1, k=2; 则逻辑表达式 ++i||--j&&++k 的值为(b)。

(a) 0　　(b) 1　　(c) 2　　(d) 3

24. 执行下列语句后，x的值是(d)，y的值是(c)。

```
int x, y ;
x = y = 1;  ++ x || ++ y ;
```

(a) 不确定　　(b) 0　　(c) 1　　(d) 2

25. 设x为整型变量,不能正确表达数学关系 1<x<5 的C++逻辑表达式是(a)。

(a) 1< x <5　　(b) x==2||x==3||x==4

(c) 1<x && x<5　　(d) !(x<=1)&&!(x>=5)

26. 已知 int x=5; 执行下列语句后，x的值为(c)。

```
x += x -= x * x;
```

(a) 25　　(b) 40　　(c) –40　　(d) 20

27. 设 int a=1, b=2, c=3, d=4; 则以下条件表达式的值为(a)。

```
a < b ? a : c < d ? c : d
```

(a) 1　　(b) 2　　(c) 3　　(d) 4

28. 以下逗号表达式的值为(d)。

```
( x = 4 * 5, x * 5 ), x + 25
```

(a) 25　　(b) 20　　(c) 100　　(d) 45

1.2 把下列数学表达式写成 c++算术表达式

1. $\cfrac{1}{1+\cfrac{1}{1+\cfrac{1}{x+y}}}$

2. x{x[x(ax+b)+c]+d}+e

3. $\ln\left(1+\left|\frac{a+b}{a-b}\right|^{10}\right)$

4. $\sqrt{1+\frac{\pi}{2}\cos 48^{\circ}}$

5. $\cot\left(\frac{1-x^2}{1+x^2}\right)$

6. $\lg(a^2+ab+b^2)$

【解答】

1. 1/(1 + 1/(1 + 1/(x + y)))

2. x * (x * (x * (a * x + b) + c) + d) + e

3. log(1 + pow(fabs((a + b)/(a - b)), 10)

4. sqrt(1 + 3.14159/2 * cos(48 * 3.14159/180))

5. 1/tan((1 - x*x)/(1 + x*x))

   或者 cos((1 - x*x)/(1 + x*x))/sin((1 - x*x)/(1 + x*x))

6. log10(a * a + a * b + b * b)

1.3 用逻辑表达式表示下列条件

1. i 被 j 整除

2. n 是小于正整数 k 的偶数

3. 1≤x<10

4. x，y 其中有一个小于 z

5. y ∉ [-100,-10]，并且 y ∉ [10,100]

6. 坐标点（x, y）落在以（10, 20）为圆心，以 35 为半径的圆内

7. 三条边 a，b 和 c 构成三角形

8. 年份 Year 能被 4 整除，但不能被 100 整除或者能被 400 整除

【解答】

1. i%j == 0

2. (n<k) && (n%2 == 0)

3. 1<=x && x<10

4. x<z||y<z

5. !(y>=-100 && y<=-10) && !(y>=10 && y<=100)

6. sqrt(pow((x-10),2) + pow((y-20),2))< 35

7. a+b>c && b+c>a && c+a>b

8. (year%4 == 0) && (year%100!=0)||(year%400==0)

1.4 阅读下列程序，写出执行结果

1.

```
#include <iostream>
using namespace std;
int main()
{ int a = 1, b = 2;
  bool x, y;
  cout << (a++)+(++b) << endl;
  cout << a % b << endl;
  x = !a>b;
  y = a-- && b;
  cout << x << endl;
  cout << y << endl;
}
```

【解答】

```
4
2
0
1
```

2.

```
#include <iostream>
using namespace std;
int main()
{ int x,y,z,f;
```

```
    x = y = z = 1;
    f = --x || y-- && z++;
    cout << "x = " << x << endl;
    cout << "y = " << y << endl;
    cout << "z = " << z << endl;
    cout << "f = " << f << endl;
  }
```

【解答】

```
x=0
y=0
z=2
f=1
```

3.

```
#include <iostream>
#include<iomanip>
using namespace std;
int main()
{ int a=123;
  int &ra=a;
  int *pa=&a;
  cout<<setw(5)<<dec<<a<<setw(5)<<oct<<ra<<setw(5)<<hex<<*pa<<endl;
}
```

【解答】

123　173　7b

1.5　思考题

1．什么叫数据类型？变量的类型定义有什么作用？

【解答】

数据“类型”是对数据的抽象。类型相同的数据有相同的表示形式、存储格式以及相关的操作。定义一个变量时，计算机根据变量的类型分配存储空间，并以该类型解释存放的数据。

2．普通数据类型变量和指针类型变量的定义、存储、使用方式上有何区别？请编写一个程序验证之。

【解答】

变量类型	定义	存储	使用方式
数据	类型 标识符	数据值	通过名访问即直接访问对变量内容操作
指针	类型 * 标识符	地址值	通过指针变量的地址值间址访问对象

验证程序：

```
#include<iostream>
using namespace std;
int main()
{ int a,b,c;
  cout<<"a,b,c= ";
  cin>>a>>b>>c;              //对普通数据类型变量赋值
  int *pa=&a,*pb=&b,*pc=&c;                          //用变量地址值初始化指针变量
  cout<<"a,b,c= "<<a<<", "<<b<<", "<<c<<endl;    //名访问，输出a,b,c的值
  cout<<"pa,pb,pc= "<<pa<<", "<<pb<<", "<<pc<<endl;  //输出指针变量的地址值
```

```
    //间址访问，输出pa,pb,pc指向的变量的赋值
    cout<<"*pa,*pb,*pc= "<<*pa<<", "<<*pb<<", "<<*pc<<endl;
}
```

3．什么叫数据对象的引用？对象的引用和对象的指针有什么区别？请用一个验证程序说明之。

【解答】

引用是为数据对象定义别名。引用与指针有以下几点区别：

（1）引用名不是内存变量，而指针变量要开辟内存空间。

（2）引用名需要在变量定义与变量名绑定，并且不能重定义；指针变量可以在程序中赋给不同的地址值，改变指向。

（3）程序中用变量名和引用名访问对象的形式和效果一样；指针变量通过间址访问对象。

验证程序：

```
#include<iostream>
using namespace std;
int main ()
{ int a;
  cout<<"a=";
  cin>>a;
  int ra=a;
  int *pa=&a;
  cout<<"a 的值："<<a<<endl;
  cout<<"a 的地址："<<&a<<endl;
  cout<<"ra 的值："<<ra<<endl;
  cout<<"ra 的地址："<<&ra<<endl;
  cout<<"pa 所指向的变量的值："<<*pa<<endl;
  cout<<"pa 的地址："<<pa<<endl;
}
```

4．数据对象在C++中有什么不同的访问方式？请编写一个程序验证之。

【解答】

数据对象在C++中的访问方式有：名访问，引用（别名）访问，间址访问。

验证程序：

```
#include<iostream>
using namespace std;
int main()
{ int a;
  cout<<"a=";
  cin>>a;
  a=a+5;                    //名访问
  cout<<&a<<endl;           //输出变量地址
  cout<<*(&a)<<endl;        //地址访问，输出变量值
  int *pa=&a;               //说明指针变量，指向变量 a
  cout<<*pa<<endl;          //间址访问，输出变量值
  int &ra=a;                //ra 是 a 的引用
  cout<<ra<<endl;           //引用访问，输出变量 a 的值
}
```

5．为了约束对数据对象的值做只读操作，C++采用什么方式？请做出简要归纳。

【解答】

约束数据对象只读形式如下：

约束对象	说明形式
标识常量	const 类型 常量标识符=常量表达式;
指针常量	类型 * const 指针;
指向常量的指针	const 类型 * 指针; 或者 类型 const * 指针;
指向常量的指针常量	const 类型 * const 指针; 或者 类型 const * const 指针;
常引用	const 类型 & 引用名 = 对象名;

6．什么叫表达式？表达式值的类型由什么因素决定？使用不同运算符连接以下3个变量，请写出5个以上获得值等于true的表达式。

```
int a=1, b=2; double x=0.5;
```

【解答】

表达式是由数据和运算符，按求值规则，表达一个值的式子。

表达式值的类型的决定因素为操作数的类型。

（1）如果运算符左右操作数类型相同，运算结果也是相同类型。

（2）如果运算符左右操作数类型不同，首先把类型较低（存储要求，示数能力较低）的数据转换成类型较高的数据，然后运算。

（3）赋值表达式的类型由被赋值变量的类型决定。当把一个表达式的值赋给一个变量时，系统首先强制把运算值转换成变量的类型，然后执行写操作。

6 个值等于 true 的表达式：

（1）b>a && a>x　　（2）(a+b)!=x　　（3）a||(b+x)

（4）a==(b*x)　　（5）a-b<x　　（6）(a/x==b)

7．阅读以下程序，分析下面语句序列中每一个字符“*”和“&”的意义，写出输出结果。

```
#include <iostream>
using namespace std;
int main()
{ int a=10, b=20;
  int *p = &a, *q = &b;
  *p = *p * *q;
  int & ra = a;
  ra=a;
  int * & rt = q;
  *rt = 30;
  cout<<"a="<<a<<"\nb="<<b<<"\n*p="<<*p<<"\n*q="<<*q
      <<"\nra="<<ra<<"\n*rt="<<*rt<<endl;
}
```

【解答】

字符“*”和“&”的意义见程序中添加的注释。

```
#include <iostream>
using namespace std;
int main()
{ int a=10, b=20;
  int *p = &a, *q = &b;      //“*”是指针类型说明符，“&”是取址运算符
```

```
    *p = *p * *q;          //第1、2、4个“*”是间址访问符，第3个“*”算术乘运算符
    int & ra = a;          //“&”是引用说明符
    ra=a;
    int * & rt = q;        //“*”是指针类型说明符，“&”是引用说明符
    *rt = 30;              //“*”是间址访问符
    //输出语句中的“*”是间址访问符
    cout<<"a="<<a<<"\nb="<<b<<"\n*p="<<*p<<"\n*q="<<*q
        <<"\nra="<<ra<<"\n*rt="<<*rt<<endl;
}
```

程序输出结果为：

```
a=200
b=30
*p=200
*q=30
ra=200
*rt=30
```

1.6 编程题

1.输入一个三位整数，将它反向输出。

【解答】

```
#include <iostream>
using namespace std;
int main()
{ int x,i,j,k;
  cout << "please input x:";
  cin >> x;
  i = x/100;
  j = x/10 %10;
  k = x%10;
  cout << k << j << i << endl;
}
```

2.输入平面上某点横坐标 x 和纵坐标 y，若该点在由图 3.1 表示的方块区域内，则输出 1；否则，输出 0。

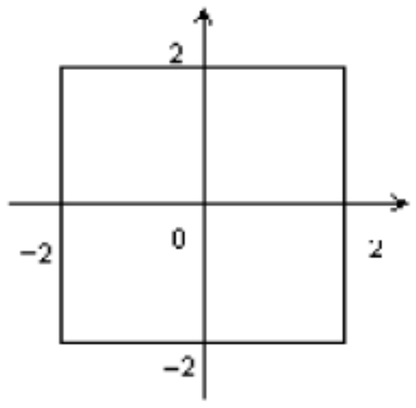


图 1.11 正方形

【解答】

```
#include <iostream>
using namespace std;
```

```
int main()
 { double x,y,b;
    cout << "please input x,y:";
    cin >> x >> y;
    b = ( -2<=x ) && ( x<=2 ) && ( -2<=y ) && ( y<=2 );
    cout << b << endl;
 }
```

3.输入三个整数，求出其中最小数（要求使用条件表达式）。

【解答】

```
#include <iostream>
using namespace std;
int main()
{ int a,b,c,temp,min;
  cout << "please input a,b,c:";
  cin >> a >> b >> c;
  temp = ( a<b ) ? a:b;
  min = ( temp<c ) ? temp:c;
  cout << "min=" << min << endl;
}
```

习题 2 及其解答

2.1 选择题

1. 已知 int i=0, x=1, y=0 ; 在下列选项使i的值变成1的语句是(c)。

(a) if(x&&y) i++ ;　　(b) if(x==y) i++ ;

(c) if(x||y) i++ ;　　(d) if(!x) i++ ;

2. 设有函数关系为 y=$\begin{cases} -1 & x<0 \\ 0 & x=0 \\ 1 & x>0 \end{cases}$，下面选项中能正确表示上述关系为（ c ）。

(a)
```
y = 1 ;
if( x >= 0 )
  if( x == 0 ) y = 0 ;
  else  y = -1;
```

(b)
```
y = -1 ;
if( x != 0 )
   if( x > 0 ) y = 1 ;
    else y = 0;
```

(c)
```
if( x <= 0 )
   if( x < 0 ) y = -1 ;
     else  y = 0 ;
else  y = 1 ;
```

(d)
```
y = -1 ;
if( x <= 0 )
   if( x < 0 ) y = -1 ;
     else y = 1 ;
```

3. 假设 i=2，执行下列语句后 i 的值为（ b ）。

```
switch( i )
{ case 1 : i ++ ;
  case 2 : i -- ;
  case 3 : ++ i ; break ;
  case 4 : -- i ;
  default : i ++ ;
}
```

(a) 1　　(b) 2　　(c) 3　　(d) 4

4. 已知 int i=0，x=0; 下面 while 语句执行时循环次数为（ d ）。

```
while( !x && i< 3 ) { x++ ; i++ ; }
```

(a) 4　　(b) 3　　(c) 2　　(d) 1

5. 已知 int i=3;下面 do_while 语句执行时循环次数为（ b ）。

```
do{ i--; cout<<i<<endl;}while ( i!= 1 ) ;
```

(a) 1　　(b) 2　　(c) 3　　(d) 无限

6. 下面 for 语句执行时循环次数为（ b ）。

```
int i;
for ( i=0, j=5;i=j; )
 { cout << i << j << endl;
   i++; j--;
 }
```

(a) 0　　(b) 5　　(c) 10　　(d) 无限

7. 以下死循环的程序段是（ b ）。

(a) int x; for(int x=0 ; x<3 ;) { x++ ; } ;

(b) int k = 0; do { ++k ; } while(k>=0) ;

(c) int a=5 ; while(a) { a-- ; } ;

(d) int i=3 ; for(; i ; i --) ;

2.2 阅读下列程序，写出执行结果

1.

```
#include<iostream>
using namespace std;
int main()
{ int a,b,c,d,x;
  a = c = 0; b = 1; d = 20;
  if( a ) d = d-10;
    else if( !b )
      if( !c )
        x = 15;
       else x = 25;
  cout << d << endl;
}
```

【解答】

20

2.

```
#include<iostream>
using namespace std;
int main()
{ int a = 0, b = 1;
  switch( a )
  { case 0: switch( b )
            { case 0 : cout<<"a="<<a<<" b="<<b<<endl; break;
              case 1 : cout<<"a="<<a<<" b="<<b<<endl; break;
            }
    case 1: a++; b++; cout<<"a="<<a<<" b="<<b<<endl;
  }
}
```

【解答】

a= 0 b= 1

a= 1 b= 2

3.

```
#include<iostream>
using namespace std;
int main()
{ int i = 1;
  while( i<=10 )
    if( ++i % 3 != 1 )
       continue;
    else cout << i << endl;
}
```

【解答】

4

7

10

4.

```
#include <iostream>
using namespace std;
 int main()
  { int i = 0 , j = 5;
    do
      { i++; j--;
        if ( i>3 ) break;
      } while ( j>0 );
    cout << "i=" << i << '\t'<< "j=" << j << endl;
  }
```

【解答】

i= 4　　j= 1

5.

```
#include<iostream>
using namespace std;
int main()
{ int i,j;
  for( i=1, j=5; i<j; i++ )
    { j--; }
  cout<<i<<'\t'<<j<<endl;
}
```

【解答】

3　　3

6.

```
#include<iostream>
using namespace std;
int main()
{ int i, s = 0;
  for( i=0; i<5; i++ )
    switch( i )
      { case 0:  s += i;  break;
        case 1:  s += i;  break;
        case 2:  s += i;  break;
        default: s += 2;
      }
  cout<<"s="<<s<<endl;
}
```

【解答】

s= 7

7.

```
#include<iostream>
```

```
using namespace std;
int main()
{ int i, j, x = 0;
  for( i=0; i<=3; i++ )
   { x++;
     for( j=0; j<=3; j++ )
      { if( j % 2 ) continue;
        x++;
      }
      x++;
    }
   cout << "x=" << x << endl;
}
```

【解答】

x= 16

2.3 思考题

1. C++中有什么形式的选择控制语句？归纳它们语法形式、应用场合。根据一个实际问题使用不同的条件语句编程。

【解答】

语句	使用方式	使用场合
if 语句	if(表达式)语句 1; else 语句 2;	需要对给定的条件进行判断，并根据判断的结果选择不同的操作。 适用于复杂的条件表达式判断。
switch 语句	switch(表达式) { case 常量表达式 1: 语句 1; case 常量表达式 2: 语句 2; …… case 常量表达式 n; 语句 n; [default : 语句 n+1;] }	根据整型表达式的不同值决定程序分支的情况。 适用于判断表达式简单，需要多个分支处理的情况。

演示程序：

程序（1）

```
//此程序用if输出等级对应的分数段
//A->=90,B-(90,80],C-(80,70] ,D-(70,60],,E-<60
#include<iostream>
using namespace std;
int main()
{ char gd;
  cout<<"Enter the grade:";
  cin>>gd;
  //直到输入有效等级，否则程序不继续运行
  while(!((gd>='A' && gd<='E')||(gd>='a' && gd<='e')))
   { cout<<"Invalid grade! Please retry:";
     cin>>gd;
```

```
    }
   if(gd=='A'||gd=='a') cout<<"\nScored 90-100!\n";
       else if(gd=='B'||gd=='b') cout<<"\nScored 80-89!\n";
             else if(gd=='C'||gd=='c') cout<<"\nScored 70-79!\n";
                   else if(gd=='D'||gd=='d') cout<<"\nScored 60-69!\n";
                         else if(gd=='E'||gd=='e') cout<<"\nScore under 60!\n";
                               else cout<<"Unexpect error!\n";  //防止意外错误
}
```

程序（2）

```
//此程序用switch输出等级对应的分数段
//A->=90,B-(90,80],C-(80,70] ,D-(70,60],,E-<60
#include<iostream>
using namespace std;
int main()
{ char gd;
  cout<<"Enter the grade:";
  cin>>gd;
  //直到输入有效等级，否则程序不继续运行
  while(!((gd>='A' && gd<='E')||(gd>='a' && gd<='e')))
   { cout<<"Invalid grade! Please retry:";
     cin>>gd;
   }
   switch(gd)
   { case 'A':
     case 'a': cout<<"\nScored 90-100!\n";break;
     case 'B':
     case 'b': cout<<"\nScored 80-89!\n";break;
     case 'C':
     case 'c':cout<<"\nScored 70-79!\n";break;
     case 'D':
     case 'd':cout<<"\nScored 60-69!\n";break;
     case 'E':
     case 'e':cout<<"\nScore under 60!\n";break;
     default:cout<<"Unexpect error!\n";//防止意外错误
   }
}
```

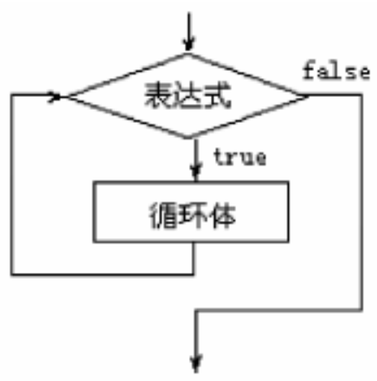
2. 什么叫循环控制？归纳比较 C++中各种循环控制语句的语法、循环条件和循环结束条件的表示形式及执行流程。

【解答】

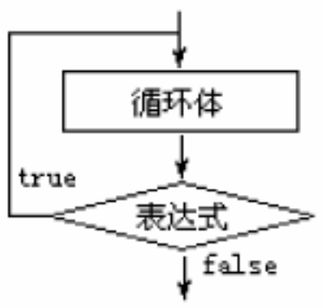
循环控制是在特定的条件下，程序重复执行一些特定动作。

语句	语法	执行流程	使用场合

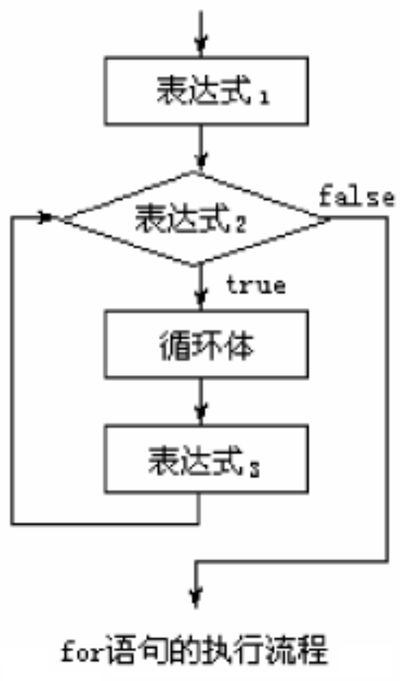
<table>
<tr>
<td>while 语句</td>
<td>while(表达式)
　循环体;

循环条件：表达式值为非 0(真)
循环结束条件：表达式值为 0(假)</td>
<td> 

while语句的执行流程</td>
<td>程序中常用于根据条件执行操作而不需关心循环次数的情况。
先判断形式循环，条件不成立时不进入循环体。</td>
</tr>
<tr>
<td>do-while
语句</td>
<td>do
循环体
while(表达式);

循环条件：表达式值为非 0(真)
循环结束条件：表达式值为 0(假)</td>
<td> 

do_while语句执行流程</td>
<td>程序中常用于根据条件执行操作而不需关心循环次数。
后判断形式循环，至少执行 1 次循环体。
一般情况，while 语句和 do while 语句可以互换使用。</td>
</tr>
<tr>
<td>for 语句</td>
<td>for([表达式 1];[表达式 2];[表达式 3])
　　循环体;

（1）表达式 1 称为初始化表达式，不是循环体执行部分。
（2）表达式 3 称为后置表达式，作为循环体的最后一个执行表达式。
（3）循环条件：表达式 2 值为非 0（真）
循环结束条件：表达式 2 值为 0（假）</td>
<td> 

for语句的执行流程</td>
<td>for 语句称为步长循环语句，通常用于确定循环次数的情况。
由于语句的 3 个表达式均可以缺省，也可以用于条件循环，即循环次数不确定的情况。</td>
</tr>
</table>

3. 根据一个实际问题，用不同的循环语句编程，分析其优缺点。

【解答】

略。

4. 用 if 语句和 goto 语句组织循环，改写思考题的第 3 小题编写的程序。分析在什么情况下可以适当使用 goto 语句。

【解答】

在不破坏程序基本流程控制的情况下，可以适当使用 goto 语句实现从语句结构内部向外的必要跳转，即按特定条件结束结构语句块的执行。

程序略。

5.有以下程序

```
#include<iostream>
using namespace std;
int main()
{ char c;
  cin>>c;
  if(c=='y'||c=='Y')
      int a=1;
  else
```

```
        int a=0;
      cout<<"a="<<a<<endl;
    }
```

编译错误为：error C2065: 'a' : undeclared identifier，指示变量 a 没有定义。请分析原因，并做出修改。

【解答】

变量a的定义不应该放在if-else语句体中。说明语句和执行语句的执行时机不同。变量说明要求在编译时定义存储空间，而if-else是执行语句，程序运行后才执行。正确的程序是：

```
#include<iostream>
using namespace std;
int main()
{ char c;
  int a;
  cin>>c;
  if(c=='y'||c=='Y')
    a=1;
  else
    a=0;
  cout<<"a="<<a<<endl;
}
```

6. 有以下程序，希望判断两个输入的整数是否相等。程序通过编译，但不能达到预期结果。请分析程序能够通过 C++编译而不能得到期望结果的原因。

```
#include<iostream>
using namespace std;
int main()
{ int a,b;
  cout<<"a: ";  cin>>a;
  cout<<"b: ";  cin>>b;
  if( a=b )
    cout<<a<<"等于"<<b<<endl;
  else
    cout<<a<<"不等于"<<b<<endl;
}
```

程序运行后，输入 a 的值为 4，b 的值为 9，显示结果如下：

```
a: 4
b: 9
9 等于 9
```

【解答】

在 if 语句的判断表达式(a=b)中，赋值号“=”应该是逻辑等“==”。从语法上，C++的 if 语句把 a=b 这个赋值表达式视为逻辑表达式，没有编译错误。a=b 的值决定于 b。若 b 的输入值不等于 0，if 作为逻辑真（true），否则作为逻辑假（false）。所以，题目中输入 b 的值虽然不等于 a，但表达式 a=b 为逻辑 true，执行了 if 语句的第 1 个分支。

2.4 编程题

1. 输入某学生成绩，若成绩在85分以上输出“very good”，若成绩在60分到85分之间输出“good”，若成绩低于60分输出“no good”。

【解答】

```
#include<iostream>
using namespace std;
int main()
{ double score;
   cout << "please input score:";
   cin >> score;
   if ( score>=85 ) cout << "Very good!" ;
      else if ( score>=60 ) cout << "Good!";
            else cout << "No good!";
}
```

2. 输入三个整数，按从小到大的顺序输出它们的值。

【解答】

```
#include<iostream>
using namespace std;
int main()
{ int a, b, c, t;
  cout << "a, b, c=";
  cin >> a >> b >> c;
  if(a>b) { t=a; a=b; b=t; }
  if(a>c) { t=a; a=c; c=t; }
  if(b>c) { t=b; b=c; c=t; }
  cout<< a << '\t'<< b << '\t' << c << endl;
}
```

3. 输入三角形的三条边，判别它们能否形成三角形，若能，则判断是等边、等腰、还是一般三角形。

【解答】

```
#include<iostream>
using namespace std;
int main()
{ double a, b, c ;
  cout << "a, b, c = " ;
  cin >> a >> b >> c ;
  if ( a+b > c && b+c > a && c+a > b )
{ if ( a == b && b == c )
     cout << "等边三角形！" << endl;
   else if ( a == b || a == c || b == c )
          cout << "等腰三角形！" << endl;
        else cout << "一般三角形！" << endl;
}
else
 cout << "不能形成三角形！" << endl ;
```

}

4．输入百分制成绩，并把它转换成五级分制，转换公式为：

$$grade(\text{级别})=\begin{cases} A(\text{优秀}) & 90-100 \\ B(\text{良好}) & 80-89 \\ C(\text{中等}) & 70-79 \\ D(\text{合格}) & 60-69 \\ E(\text{不合格}) & 0-59 \end{cases}$$

【解答】

```
#include<iostream>
using namespace std;
int main()
{ double score; char grade;
   cout << "score=";
    cin >> score;
   if ( score >= 0 && score <= 100 )
     { switch  ( int( score ) /10 )
          { case   10:
            case   9:  grade = 'a'; break;
            case   8:  grade = 'b'; break;
            case   7:  grade = 'c'; break;
            case   6:  grade = 'd'; break;
            case   5:
            case   4:
            case   3:
            case   2:
            case   1:
            case   0:  grade = 'e'; break;
          }
      }
    else
     { cout <<"数据输入错误！"<< endl;
       goto end;
      }
       cout << grade << endl;
  end:  ;                          //分号不能省
}
```

5．编程序模拟剪刀、石头和纸游戏。游戏规则为：剪刀剪纸，石头砸剪刀，纸包石头。玩游戏者从键盘上输入 s（表示剪刀）或 r（表示石头）或 p（表示纸），要求两个游戏者交替输入，计算机给出输赢的信息。

【解答】

```
#include<iostream>
using namespace std;
int main()
{ char first,second;
```

```
cout << "First input( s,r or p ):";
cin >> first;
cout << "Second input( s,r or p ):";
cin >> second;
switch  ( first )
{ case  's':
     switch  ( second )
       { case  's': cout << "Scissor ties scissor." << endl; goto end;
         case  'r': cout << "Scissor is crushed by rock." << endl; goto end;
         case  'p': cout << "Scissor cuts paper." << endl; goto end;
         default :  cout << "second input error!" << endl ; goto end;
      }
case  'r':
    switch  ( second )
     { case  's': cout << "Rock crushes scissor." << endl; goto end;
       case  'r': cout << "Rock ties rock." << endl; goto end;
       case  'p': cout << "Rock is wrapped by paper." << endl; goto end;
       default :  cout << "second input error!" << endl; goto end;
     }
case  'p':
     switch  ( second )
      { case  's': cout << "Paper is cut by scissor." << endl; goto end;
        case  'r': cout << "Paper wraps the rock." << endl; goto end;
        case  'p': cout << "Paper ties paper." << endl; goto end;
        default :  cout << "second input error!" << endl; goto end;
      }
   default : cout << "First input error!" << endl; goto end;
  }
  end: ;
}
```

6.输入一个整数，输出该整数的所有素数因子。例如，输入 120，输出为 2、2、2、3 和 5。

【解答】

```
#include<iostream>
using namespace std;
int main()
{ int m,i = 2;
cout << "please input m:";
cin >> m;
while( i<=m )
  if( m % i == 0 )
   { cout << i << ",";
     m = m / i;
   }
 else i++;
```

```
}
```

7. 使用迭代公式 $x_{n+1}=(x_n+a/x_n)/2 \quad (n=0,1,2,\cdots;x_0=a/2)$ 编程序求某一正整数 a 的平方根。

【解答】

```
#include<iostream>
#include<cmath>
using namespace std;
int main()
{ const double eps = 1e-8;
  double a,x0,x;
   cout << "please input a:";
   cin >> a;
   x0 = a / 2;
   x = ( x0 + a/x0 )/2;
   while( fabs( x-x0 )>eps )
    {  x0 = x;  x =( x0 + a/x0 )/2;
    }
  cout << x << endl;
}
```

8. 已知 X=0°，10°，20°，…，180°，求 sin$x$,cos$x$ 和 tan$x$ 的值。

【解答】

```
#include<iostream>
#include<cmath>
#include<iomanip>
using namespace std;
int main()
{ const double pi = 3.14159265;
  int i;
  double x,y1,y2,y3;
  cout << setw(2) << "x" << setw(15) << "sin(x)" << setw(15)
       << "cos(x)" << setw(15) << "tg(x)" << endl;
  for( i=0; i<=18; i++ )
    { x = i*10*pi/180;
      y1 = sin( x );
      y2 = cos(x);
      y3 = y1/y2;
      cout << setw(2) << i << setw(15) << y1 << setw(15)
           << y2 << setw(15) << y3 << endl;
    }
}
```

9. 在 100 到 200 中找出同时满足用 3 除余 2，用 5 除余 3 和用 7 除余 2 的所有整数。

【解答】

```
#include<iostream>
using namespace std;
```

```
int main()
{ int i;
  for( i=100; i<=200; i++ )
   { if ( ( i % 3 == 2) && ( i % 5 == 3 ) && ( i % 7 == 2 ) )
       cout << i << endl;
   }
}
```

10.求 100 到 999 中的水仙花数。所谓水仙花数是指一个三位数，它的每位数字的立方之和等于该数。例如，因为 $153=1^3+5^3+3^3$，所以 153 为水仙花数。

【解答】

```
#include<iostream>
using namespace std;
int main()
{ int i,a,b,c;
   for( i=100; i<=999; i++ )
     {  a = i/100;
        b = ( i-a*100 ) / 10;
        c = i - a*100 - b*10;
        if ( i == a*a*a + b*b*b + c*c*c ) cout << i <<endl;
     }
}
```

11.求 1000 之内的所有完数。所谓完数是指一个数恰好等于它的所有因子之和。例如，因为 6=1+2+3，所以 6 为完数。

【解答】

```
#include<iostream>
using namespace std;
int main()
{ int i,j,s;
    for( i=1; i<=1000; i++ )
     { s = 0;
       for( j=1; j<i; j++ )
         if ( i % j == 0 ) s = s + j;
       if ( i == s ) cout << i << endl;
     }
}
```

12.编一程序显示由符号组成的三角形图案。例如，程序运行后，

屏幕显示：　How many lines ?

用户输入：　5

屏幕显示：　What character ?

用户输入：　*

则输出如下图案。

```
    *
   * * *
```

```
        *****
       *******
      *********
```

【解答】

```
#include<iostream>
using namespace std;
int main()
{ int i,j,k,n;
  char ch;
  cout<<"How many lines ?\n";
  cin>>n;
  cout<<"What character ?\n";
  cin>>ch;
  for( i=1; i<=n; i++ )
   {  for( k=1; k<=n-i; k++ ) cout << " ";
      for( j=1; j<=2*i-1; j++ ) cout << ch ;
      cout << endl;
   }
}
```

13.已知 XYZ+YZZ=532，其中 X，Y 和 Z 为数字，编一程序求出 X，Y 和 Z 的值。

【解答】

```
#include<iostream>
using namespace std;
int main()
{ int x,y,z,i;
  for( x=1; x<=9; x++ )
    for( y=1; y<=9; y++ )
      for( z=0; z<=9; z++ )
       {  i = 100*x + 10*y + z + 100*y + 10*z + z;
          if ( i == 532 )
             cout<<"x="<<x<<'\t'<<"y="<<y<<'\t'<<"z="<<z<<endl;
       }
}
```

习题 3 及其解答

3.1　选择题

1. 以下正确的函数原型为(　d　)。

(a) f1(int x; int y);　　(b) void f1(x, y);

(c) void f1(int x, y);　　(d) void f1(int, int);

2. 有函数原型 void fun2(int); 下面选项中，不正确的调用是(　c　)。

(a) int a = 21;　fun2(a);　　(b) int a = 15;　fun2(a*3);

(c) int b = 100; fun2(&b);　　(d) fun2(256);

3. 有函数原型 void fun3(int *); 下面选项中，正确的调用是(　c　)。

(a) double x = 2.17; fun3(&x);　　(b) int a = 15 ;　fun3(a*3.14);

(c) int b = 100;　fun3(&b);　　(d) fun3(256);

4. 有函数原型 void fun4(int &); 下面选项中，正确的调用是(　c　)。

(a) int a = 2.17; fun4(&a);　　(b) int a = 15;　fun4(a*3.14);

(c) int b = 100;　fun4(b);　　(d) fun4(256) ;

5. 有声明

void fun5(int * &);　int a , *p = &a;

下面选项中，正确的调用是(　b　)。

(a) fun5(&a);　　(b) fun5(p);　　(c) fun5(*a);　　(d) fun5(*p) ;

6. 有声明

int fun6(int),(*pf)(int) = fun6;

下面选项中，正确的调用是(　c　)。

(a) int a=15; int n=fun6(&a);　　(b) int a = 15; cout<<(&pf)(a);

(c) cout<<(*pf)(256);　　(d) cout << *pf(256);

7. 在VC中，若定义一个函数的返回类型为void，以下叙述正确的是(　c　)。

(a) 函数返回值需要强类型转换　　(b) 函数不执行任何操作

(c) 函数本身没有返回值　　(d) 函数不能修改实际参数的值

8. 函数参数的默认值不允许为(　c　)。

(a) 全局常量　　(b) 直接常量　　(c) 局部变量　　(d) 函数调用

9. 使用重载函数编程序的目的是(　a　)。

(a) 使用相同的函数名调用功能相似的函数　　(b) 共享程序代码

(c) 提高程序的运行速度　　(d) 节省存贮空间

10. 下列的描述中(b)是错误的。

(a) 使用全局变量可以从被调用函数中获取多个操作结果

(b) 局部变量可以初始化，若不初始化，则系统默认它的值为0

(c) 当函数调用完后，静态局部变量的值不会消失

(d) 全局变量若不初始化，则系统默认它的值为0

11. 下列选项中，(c)的具有文件作用域。

(a) 语句标号　　(b) 局部变量　　(c) 全局变量　　(d) 静态变量

3.2 阅读下列程序，写出执行结果

1.

```
#include<iostream>
using namespace std;
#include<cmath>
int f( int ) ;
int main()
{ int i;
  for( i = 0; i < 3; i ++ )
    cout << f( i ) << endl;
}
int f( int a )
{ int b = 0 , c = 1;
  b ++; c++;
  return int( a + pow( double(b), 2 ) + c );
}
```

【解答】

3

4

5

2.

```
#include<iostream>
using namespace std;
void func(int a, int b, int c = 3, int d = 4 );
int main()
{ func( 10, 15, 20, 30 );
  func( 10, 11, 12 );
  func( 12, 12 );
}
void func( int a, int b, int c, int d )
{ cout<<a<<'\t'<<b<<'\t'<<c<<'\t'<<d<< endl;
}
```

【解答】

10	15	20	30
10	11	12	4

12　　12　　3　　4

3.

```
#include<iostream>
using namespace std;
void func( int, int, int * ) ;
int main()
{ int x, y, z;
  func( 5, 6, &x );
  func( 7, x, &y );
  func( x, y, &z );
  cout << x << " ," << y << ", "<< z << endl;
}
void func( int a , int b , int *c )
{ b += a ;  *c = b - a ;
}
```

【解答】

6, 6, 6

4.

```
#include<iostream>
using namespace std;
void func( int, int, int & );
int main()
{ int x=0 , y=1 , z=2;
  func( 1 , 2 , x );
  func( x + y , y , y );
  func( z , x + y , z );
  cout << x << " ," << y << ", "<< z << endl ;
}
void func( int a , int b , int &c )
{ b += a ;  c = b - a ;
}
```

【解答】

2, 1, 3

5.

```
#include<iostream>
using namespace std;
void func( int *, int *, int *& ) ;
int main()
{ int a=10, b=20;
  int *p=&a, *q=&b;
  func( p, q, p );
  cout << "*p=" << *p << " ,*q=" << *q << endl;
}
void func( int *t1 , int *t2 , int *& rt )
```

```
{ *t1 += 5 ;  *t2 += 5 ;
  rt = t1 ;
  *rt += 5 ;
  cout << "*t1=" << *t1 << " ,*t2=" << *t2<< " ,*rt=" << *rt << endl;
}
```

【解答】

*t1=20, *t2=25, *rt=20

*p=20, *q=25

6.

```
#include<iostream>
using namespace std;
int f2( int, int );
int f1( int a , int b )
{ int c ;
  a += a ; b += b ;
  c = f2( a+b , b+1 );
  return c;
}
int f2( int a , int b )
{ int c ;
  c = b % 2 ;
  return a + c;
}
int main()
{ int a = 3 , b = 4;
  cout << f1( a , b ) << endl;
}
```

【解答】

15

7.

```
#include<iostream>
using namespace std;
int age( int n )
{ int f;
  if( n == 1 ) f = 10 ;
    else f = age( n-1 ) + 2;
  return f ;
}
int main()
{ cout << "age : " << age( 5 ) << endl;
}
```

【解答】

age:18

8.

```
#include<iostream>
using namespace std;
int f1( int a, int b ) { return a + b ; }
int f2( int a, int b ) { return a - b ; }
int f3( int( *t )( int, int ), int a, int b ) { return ( *t )( a, b ) ; }
int main()
{ int ( *p )( int, int );
  p = f1 ;
  cout << f3( p, 4, 8 ) << endl;
  p = f2 ;
  cout << f3( p, 8, 4 ) << endl;
}
```

【解答】

```
12
4
```

9.

```
#include<iostream>
using namespace std;
int sub( int, int );
int a = 1 ;
int main()
{ int m = 1, n = 2, f;
  f = sub( m, n );
  cout << a << '\t' << f << endl;
  f = sub( m, n ) ;
  cout << a << '\t' << f << endl;
}
int sub( int c, int d )
{ static int m = 2, n = 5 ;
  cout << m << '\t' << n << '\t' << endl;
  a = ++ a ; c = m ++ ; d = n ++;
  return c + d ;
}
```

【解答】

```
2       5
2       7
3       6
3       9
```

3.3 思考题

1．函数的作用是什么？如何定义函数？什么叫函数原型？

【解答】

函数的两个重要作用：

（1）任务划分，把一个复杂任务划分为若干小任务，便于分工处理和验证程序正确性；（2）软件重用，把一些功能相同或相近的程序段，独立编写成函数，让应用程序随时调用，而不需要编写雷同的代码。

函数的定义形式：

```
类型　函数名 （［ 形式参数表 ］）
{
　　语句序列
}
```

函数原型是函数声明，告诉编译器函数的接口信息：函数名、返回数据类型、接收的参数个数、参数类型和参数顺序，编译器根据函数原型检查函数调用的正确性。

2．什么叫函数值的返回类型？什么叫函数的类型？如何通过指向函数的指针调用一个已经定义的函数？请写一个验证程序说明。

【解答】

（1）函数的返回类型是函数返回的表达式的值得类型；

（2）函数类型是指函数的接口，包括函数的参数定义和返回类型；

（3）若有

```
functionType functionName;  //functionType 是已经定义的函数类型
functionType *functionPointer=functionName;    //定义函数指针并获取函数地址
```

则可以通过函数指针调用函数：

```
(*functionPointer)(argumentList);
```

或　`functionPointer(argumentList);`

其中 argumentList 是实际参数表。

验证程序：

```
#include<iostream>
using namespace std;
int main()
{ typedef int myfunc(int,int);
  myfunc f,*fp;
  int a=10,b=6;
  fp=f;
  cout<<"Using f(a):"<<f(a,b)<<endl;          //函数名调用函数
  cout<<"Using fp(a):"<<fp(a,b)<<endl;        //函数指针调用函数
  cout<<"Using (*fp)(a):"<<(*fp)(a,b)<<endl;    //函数指针调用函数
  return 0;
}
int f(int i,int j)
{ return i*j;
}
```

3．什么叫形式参数？什么叫实际参数？C++函数参数有什么不同的传递方式？请写一个验证程序说明。

【解答】

参数是调用函数与被调用函数之间交换数据的通道。函数定义首部的参数称为形式参数，调用函数时使用的参数称为实际参数。C++有三种参数传递机制：值传递（值调用）；指针传递（地址调用）；引用传递（引用调用）。

验证程序：

```
#include<iostream>
using namespace std;
```

```
void funcA(int i)
{ i=i+10; }
void funcB(int *j)
{ *j=*j+20; }
void funcC(int &k)
{ k=k+30; }
int main()
{ int a=1;
  funcA(a);cout<<"a="<<a<<endl;
  funcB(&a);cout<<"a="<<a<<endl;
  funcC(a);cout<<"a="<<a<<endl;
}
```

程序输出：

```
a=1        //传值参数，实际参数值不变
a=21       //指针参数，形式参数通过间址修改实际参数
a=51       //引用参数，形式参数通过别名方式修改实际参数
```

4．C++函数通过什么方式传递返回值？当一个函数返回指针类型时，对返回表达式有什么要求？若返回引用类型时，是否可以返回一个算术表达式？为什么？

【解答】

C++首先计算表达式的值，然后把该值赋给函数返回类型的匿名对象，通过这个对象，把数值带回调用点，继续执行后续代码。

当函数返回指针类型时，返回的地址值所指对象不能是局部变量。因为局部变量在函数运行结束后会被销毁，返回这个指针是毫无意义的。

返回引用的对象不能是局部变量，也不能返回表达式。算术表达式的值被储存在匿名空间中，函数运行结束后会被销毁，返回这个变量的引用也是无意义的。

5．变量的生存期和变量作用域有什么区别？请举例说明。

【解答】

变量的生存期是指程序运行后，变量占有内存的时间；变量作用域指的是指变量声明之后，在程序正文中有效的那部分区域。

例如，定义函数:

```
void count()
{ static int n=0;
  //.....
}
```

该函数中 n 被定义为 static 变量，生存期是整个程序运行时期；但作用域只在 count 函数中。

6．静态局部变量有什么特点？编写一个应用程序，说明静态局部变量的作用。

【解答】

静态局部变量的生存期是全程的，作用域是局部的。程序开始执行时就分配和初始化存储空间（默认初始化值为0）。定义静态局部变量的函数退出时，系统保持其存储空间和数值。下次调用这个函数时，static变量还是上次退出函数时的值。直至整个程序运行结束，系统才收回存储空间。

程序略。

7．在一个语句块中能否访问一个外层的同名局部变量？能否访问一个同名的全局变量？ 如果可以，应该如何访问？请写一个验证程序说明。

【解答】

一个语句块中不能访问外层的同名局部变量。可以访问一个同名的全局变量。

验证程序：

```
#include<iostream>
using namespace std;
int a=0;          //全局变量a
int main()
{ int a=1;        //外层局部变量a
  { int a=2;          //内层局部变量a
    cout<<"Local a is "<<a<<endl;      //输出内层局部变量a
    }
    cout<<"Main a is "<<a<<endl;          //输出外层局部变量a
    cout<<"Global a is "<<::a<<endl;      //输出全局部变量a
}
```

8. 有函数原型

```
void f (int & n) ;
```

和函数调用

```
int a;
//.....
f(a);
```

有人说，因为 n 是 a 的引用，在函数 f 中访问 n 相当于访问 a，所以，可以在 f 的函数体内直接使用变量名 a。这种说法正确吗？为什么？请你写个验证程序。

【解答】

形式参数 n 的作用域是 f 函数，实际参数 a 的作用域是调用 f 函数的模块（例如 main 函数），所以在 f 函数中可见 n 而不可见 a。因此，这种说法不正确。f 函数内不能直接使用变量名 a，只能通过别名 n 访问 a。

验证程序：

```
#include<iostream>
using namespace std;
void f ( int&n );
int main()
{ int a = 1 ;
  f( a );
  cout<<"a="<<a<<endl;
}
void f ( int&n )
{  a++;          //错误，直接使用a
   n++;          //正确
}
```

产生编译错误：error C2065: “a”: 未声明的标识符

9. 有函数原型

```
double function(int,double);
```

函数 function 的返回值类型是什么？函数的类型是什么？请用 typedef 定义函数的类型。

若有函数调用语句

```
x=function(10,(2*(0.314+5));
```

其中的括号“()”与函数原型中括号有什么语义区别？

【解答】

函数 function 的返回值类型是 double

函数类型是： double (int,double)

可以定义为： typedef double funType (int,double);

函数调用 function(10,(2*(0.314+5)) 中，外层括号表示调用函数匹配的实际参数表，里面的两层括号是表达式运算。

10. 请分析以下各语句的意义。

```
int * fun() ;
int * (*pf)() ;
fun() ;
pf = fun ;
pf() ;
```

【解答】

```
int * fun() ;          //函数原型声明。fun 是返回 int*类型，没有参数的函数
int * (*pf)();         //声明指针变量。pf 是指向函数的指针，函数类型为 int*()
fun() ;                //调用函数语句
pf = fun ;             //向指针变量赋值。函数指针 pf 指向函数 fun
pf() ;                 //用指针变量间址调用函数
```

3.4 编程题

1. 已知 $y=\dfrac{sh(1+shx)}{sh2x+sh3x}$ ， 其中 sh 为双曲正弦函数，即 $sh(t)=\dfrac{e^t-e^{-t}}{2}$。编一程序，输入 x 的值，求 y 的值。

【解答】

```
#include<iostream>
#include<cmath>
using namespace std;
double sh( double t );
int main()
{ double x,y;
  cout << "x=";
  cin >> x;
  y = sh( 1+sh(x) )/( sh( 2*x )+sh( 3*x ) );
  cout << "y=" << y << endl;
}
double sh( double t )
{  return ( exp( t )-exp( -t ) )/2; }
```

2. 输入 m、n 和 p 的值，求 s = $\dfrac{1+2+\cdots+m+1^3+2^3+\cdots+n^3}{1^5+2^5+\cdots+p^5}$ 的值。注意判断运算中的溢出。

【解答】

```
using namespace std;
double f( long k,long num );
int main()
{ long m,n,p;  double s,f1,f2,f3;
```

```
  cout << "m,n,p=";
  cin>>m>>n>>p;
  f1=f( 1,m );  f2=f( 3,n );  f3=f( 5,p );
  if (f1 && f2 && f3 )
  { s = ( f1 + f2) /f3;
    cout << "s=" << s << endl;
  }
  else cout<<"溢出!\n";
}
double f( long k,long num )
{ long i;
  double sum=0;
  for( i=1; i<=num && sum<2147483647; i++ )
    { sum = sum + pow( double (i),double (k) );
    }
  if (i<=num)
     return 0;   //溢出时返回
  return sum;
}
```

3．输入 a,b 和 c 的值，编写一个程序求这三个数的最大值和最小值。要求把求最大值和最小值编写成一个函数，并使用指针或引用作为形式参数把结果返回 main 函数。

【解答】

(1) 使用指针参数

```
#include<iostream>
using namespace std;
void fmaxmin( double,double ,double ,double *,double * );
int main()
{ double a,b,c,max,min;
  cout << "a,b,c = ";
  cin >> a >> b >> c;
  fmaxmin( a,b,c,&max,&min );
  cout << "max=" << max << endl;
  cout << "min=" <<min << endl;
}
void fmaxmin( double x,double y,double z,double *p1,double *p2 )
{ double u,v;
  if ( x>y ) { u = x; v = y; }
     else { u = y; v = x; };
  if ( z>u ) u = z;
  if ( z<v ) v = z;
  *p1 = u;
  *p2 = v;
}
```

(2) 使用引用参数

```
#include<iostream>
using namespace std;
void fmaxmin( double,double ,double ,double& ,double& );
int main()
{  double a,b,c,max,min;
   cout << "a,b,c=";
   cin >> a >> b >> c;
   fmaxmin( a,b,c,max,min );
   cout << "max=" << max << endl;
   cout << "min=" << min << endl;
}
void fmaxmin( double x,double y,double z,double &p1,double &p2 )
{ double u,v;
  if ( x>y ) { u = x; v = y; }
     else  { u = y; v = x; };
  if ( z>u ) u = z;
  if ( z<v ) v = z;
  p1 = u;
  p2 = v;
}
```

4. 用线性同余法生成随机数序列的公式为：

r_k = (multiplier * r_{k-1} + increment) % modulus

序列中的每一个数 r_k，可以由它的前一个数 r_{k-1} 计算出来。例如，如果有：

r_k = (25173 * r_{k-1} + 13849) % 65536

则可以产生 65536 个各不相同的整型随机数。设计一个函数作随机数生成器，生成一位或两位数的随机数。

利用这个随机数生成器，编写一个小学生四则运算的练习程序：

① • 可以进行难度选择。一级难度只用一位数，二级难度用两位数；

② • 可以选择运算类型，包括加、减、乘、除等；

③ • 给出错误提示；

④ • 可以统计成绩。

【解答】

```
#include<iostream>
using namespace std;
int Rand(int,int);              //生成指定范围的随机数
int main()
{ int w,i,r,t = 0;
char op,answer;
int a,b,d;
while(1)                        //练习开始
{ cout<<"现在开始？( Y 或 N )\n" ;
  cin>>answer;
  if (answer=='N'||answer=='n') break;
  while(1)
    { cout << "请输入难度( 1或2 ):";
```

```
    cin >> w;
    if ( w != 1 && w != 2 )
       cout << "输入难度错误，重新输入！" << endl;
     else break ;
  }
while(1)
  { cout << "请输入运算类型( +,-,*,/ ):" ;
    cin >> op;
    if ( op != '+' && op != '-' && op != '*' && op != '/' )
       cout << "输入运算符错误，重新输入！" << endl;
      else break;
   }
//出10道题，每题10分
t=0;
for( i=1; i<=10; i++ )
{ while(1)
   { if( w == 1 )
        { a = Rand(0,10); b = Rand(0,10);
        }
      else
        if( w == 2 )
        { a = Rand(10,100); b = Rand(10,100);
        }
    if ( op == '-' )
      if ( a<b ) continue ;                          //使被减数大于减数
    if ( op == '/' )
      if ( int( a/b ) != (a / b) ) continue;      //只做结果为整数的除法
    break;
 }
cout << a << op << b << '=';
cin >> d;
switch ( op )
  { case '+': r = a + b; break;
    case '-': r = a - b; break;
    case '*': r = a * b; break;
    case '/': r = a / b; break;
     }
   if ( r == d )
     { cout << "你算对了，加10分！" << endl;
       t = t + 10;
     }
    else cout << "你算错了！" << endl;
}
cout << "你的成绩是：" << t << "分" << endl;
```

```
  }
}
  int Rand(int m, int n)
  { static int r;          //静态变量保留上一个随机数
    do
      { r = ( 25173*r + 13849 ) % 65536 ;
      } while (r<m||r>=n);
    return r;
  }
```

5. 已知勒让德多项式为

$$p_n(x)=\begin{cases}1 & n=0\\ x & n=1\\ ((2n-1)p_{n-1}(x)-(n-1)p_{n-2}(x))/n & n>1\end{cases}$$

编一程序，从键盘上输入 x 和 n 的值，使用递归函数求 $p_n(x)$ 的值。

【解答】

```
#include<iostream>
using namespace std;
double p( double x,int n );
int main()
{ int n;
  double x;
  cout << "please input x and n:";
  cin >> x >> n;
  cout << "p(" << x << "," << n << ")=" << p( x,n ) <<endl;
}
double p( double x,int n )
{ double t1,t2;
  if( n == 0 ) return 1;
   else if( n == 1 ) return x;
          else
              { t1 = ( 2*n-1 )*p( x,n-1 );
                t2 = ( n-1 )*p( x,n-2 );
                return ( t1-t2 )/n;
              }
}
```

6. 把以下程序中的 print()函数改写为等价的递归函数。

```
#include <iostream>
using namespace std;
 void print( int w )
 { for( int i = 1 ; i <= w ; i ++ )
   { for( int j = 1 ; j <= i ; j ++ )
       cout << i << " " ;
     cout << endl ;
```

```
    }
  }
  int main()
  { print( 5 ) ;
  }
```

运行显示：

```
1
2  2
3  3  3
4  4  4  4
5  5  5  5  5
```

【解答】

```
#include<iostream>
using namespace std;
 void print(int w)
 { int i;
   if( w )
      { print( w-1 );
        for( i=1; i<=w; i++ )
           cout << w << "  ";
        cout << endl;
       }
}
void main()
{  print( 5 );
}
```

7．已知用梯形法求积分的公式为：$T_n=\frac{h(f(a)+f(b))}{2}+h\sum_{i=1}^{n-1}f(a+ih)$，其中 h =（ b—a ）/ n，n 为积分区间的等分数，编程序求如下积分的值。要求把求积分公式编写成一个函数，并使用函数指针作为形式参数。调用该函数时，给定不同的被积函数作为实际参数求不同的积分。

① $\int_0^1\frac{4}{1+x^2}dx$　　② $\int_1^2\sqrt{1+x^2}dx$　　③ $\int_0^{\pi/2}\sin x dx$

【解答】

```
#include<iostream>
#include<cmath>
using namespace std;
double f1( double x )
 {  return 4 / ( 1 + x*x );
 }
double f2( double x )
 {  return sqrt( 1 + x*x );
 }
double f3( double x )
```

```
 {  return sin( x );
 }
double trap( double( *fun )( double x ), double a,double b,long n )
 { double t,h;  int i;
   t = ( ( *fun )(a) + ( *fun )( b ) ) / 2.0;
   h = ( b - a ) / n;
   for( i=1; i<=n-1; i++ ) t += ( *fun )( a + i * h );
   t *= h;
   return t;
  }
int main()
{ double t1,t2,t3;
  t1 = trap( f1,0,1,10000 );
  cout << "t1=" << t1 << endl;
  t2 = trap( f2,1,2,10000 );
  cout << "t2=" << t2 << endl;
  t3 = trap( sin,0,3.14159265/2,10000 );
  cout << "t3=" << t3 << endl;
}
```

8. 编写一个程序，包含三个重载的 display 函数和一个主函数。要求第一个函数输出 double 值，前面用字符串“a double:”引导，第二个函数输出一个 int 值，前面用字符串“a int:”引导，第三个函数输出一个 char 字符值，前面用字符串“a char:”引导，在主函数中分别用 double、int 和 char 型变量作为实参调用 display 函数。

【解答】

```
#include<iostream>
using namespace std;
void display( double  d )
{  cout << "a double:" << d << endl;
}
void display( int i )
{  cout << "a int:" << i << endl;
}
void display( char c )
{  cout << "a char:" << c << endl;
}
int main()
{ double d = 1.5; int i = 100; char c = 'a';
  display( d );
  display( i );
  display( c );
}
```

9. 使用重载函数编程序分别把两个数和三个数从大到小排列。

【解答】

```
#include<iostream>
```

```
using namespace std;
void sort( double x,double y );
void sort( double x,double y,double z );
int main()
{ sort( 5.6, 79 );
  sort( 0.5, 30.8, 5.9 );
}
void sort(double x,double y)
{ if ( x>y ) cout << x << '\t' << y << endl;
    else cout << y << '\t' << x << endl;
}
void sort( double x,double y,double z )
{ double t;
  if( y<z ) { t = y; y = z; z = t; }
  if( x<z ) { t = x; x = z; z = t; }
  if( x<y ) { t = x; x = y ;y = t; }
  cout << x << '\t' << y << '\t' << z << '\t' << endl;
}
```

10. 给定求组合数公式为：$c_m^n = \dfrac{m!}{n!(m-n)!}$，编一程序，输入 m 和 n 的值，求 c_m^n 的值。注意优化算法，降低溢出可能。要求主函数调用以下函数求组合数：

```
int Fabricate( int m, int n ) ;        //返回 c_m^n 的值
```

Fabricate 函数内又须调用 Multi 函数：

```
int Multi( int m, int n ) ;            // 返回 m×m-1×...× n
```

程序由 4 个文件组成。头文件存放函数原型作为调用接口；其他 3 个 cpp 文件分别是 main、Fabricate 和 Multi 函数的定义。

【解答】

```
//Fabricate.h
#ifndef FABRICATE_H
#define FABRICATE_H
  int Fabricate( int m,int n );
  int Multi( int m, int n );
#endif

//main.cpp
#include<iostream>
using namespace std;
#include "Fabricate.h"
int main()
{ int m ,n;
  cout << "input m and n:";
  cin >> m >> n;
  cout << "Fabricate(" << m << "," << n << ")=" << Fabricate( m, n ) << endl;
```

```
}

//Fabricate.cpp
#include "Fabricate.h"
int Fabricate( int m, int n )
{ return Multi( m, m-n + 1 ) / Multi( n, 1 );
}

//Multi.cpp
int Multi( int m, int n )
{ int i, t = 1;
  for( i=n; i<=m; i++ )
  t = t * i;
  return t;
}
```

习题 4 及其解答

4.1　选择题

1．以下对一维数组 a 的正确定义是(　c　)。

(a) int n = 5, a[n];　　(b) int a(5);

(c) const int N = 5; int a[N];　　(d) int n; cin>>n; int a[n];

2．下列数组定义语句中，不合法的是(a　)。

(a) int a[3] = { 0, 1, 2, 3 };　　(b) int a[] = { 0, 1, 2 };

(c) int a[3] = { 0, 1, 2 };　　(d) int a[3] = { 0 };

3．已知 int a[10] = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 }, *p = a ;则不能表示数组 a 中元素的式子是(　c　)。

(a) *a　　(b) *p　　(c) a　　(d) a[p-a]

4．已知 int a[] = { 0,2,4,6,8,10 }, *p = a+1; 其值等于0的表达式是(　d　)。

(a) *(p++)　　(b) *(++p)　　(c) *(p--)　　(d) *(--p)

5．以下不能对二维数组a进行正确初始化的语句是(　c　)。

(a) int a[2][3] = { 0 };

(b) int a[][3] = { { 0,1 }, { 0 } };

(c) int a[2][3] = { { 0, 1 }, { 2, 3 }, { 4, 5 } };

(d) int a[][3] = { 0, 1, 2, 3, 4, 5 };

6．已知int a[][3] = { { 0, 1 }, { 2, 3, 4 }, { 5, 6 }, { 7 } }; 则 a[2][1]的值是(　c　)。

(a) 0　　(b) 2　　(c) 6　　(d) 7

7．已知int a[3][3] = { 1, 2, 3, 4, 5, 6, 7, 8, 9 } ; 则不能表示数组元素a[2][1]的地址是(　b)。

(a) &a[2][1]　　(b) *(a[2]+1)　　(c) a[2]+1　　(d) *(a+2)+1

8．已知char *a[]={ "fortran", " basic", "pascal", "java", "c++" } ; 则 cout<<a[3];的显示结果是(　c　)。

(a) t　　(b) 一个地址值　　(c) java　　(d) javac++

9．若用数组名作为调用函数的实参,则传递给形参的是(　a　)。

(a) 数组存贮首地址　　(b) 数组的第一个元素值

(c) 数组中全部元素的值　　(d) 数组元素的个数

10．设有char *s1="ABCDE", *s2="ABCDE",*s3=s1;下列表达式中值等于true的是(　d　)。

(a) s1==s2　　(b) s1==s3　　(c) s2==s3　　(d) strcmp(s1,s3)==0

11．设char *s1,*s2; 分别指向两个字符串。判断字符串s1和s2是否相等的表达式为(　d　)。

(a) s1=s2　　(b) s1==s2

(c) strcpy(s1,s2)==0　　(d) strcmp(s1,s2)==0

12．设char *s1,*s2; 分别指向两个字符串，判断字符串s1是否大于字符串s2的表达式为(　c　)。

(a) s1>s2　　(b) strcmp(s1,s2)==0

(c) strcmp(s1,s2)>0　　(d) strcmp(s2,s1)>0

4.2　阅读下列程序，写出执行结果

1.

```
#include <iostream>
using namespace std;
int main()
{ int i, count=0, sum=0 ;
  double average ;
  int a[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 } ;
  for( i=0; i<10; i++ )
  { if( a[i] % 2 == 0 )
      continue ;
    sum += a[ i ] ;
    count ++ ;
  }
  average = sum/count ;
  cout << "count = " << count << '\t' << "average = " << average << endl ;
}
```

【解答】

conut = 5　　average = 5

2.

```
#include <iostream>
using namespace std;
int main()
{ int a[9] = { 1, 2, 3, 4, 5, 6, 7, 8, 9 } ;
  int *p = a , sum = 0 ;
  for( ; p<a+9; p++ )
    if( *p % 2 == 0 )  sum += *p ;
  cout << "sum = " << sum << endl ;
}
```

【解答】

sum = 20

3.

```
#include <iostream>
#include <iomanip>
using namespace std;
const int N = 5 ;
int main()
{ int a[N][N]={ 0 }, i, j, k ;
  for( k=1 , i=0 ; i<N ; i++ )
    for( j=i; j>= 0; j-- , k++ )
      a[j][i-j] = k ;
  for( i=0 ; i<N ; i++ )
    { for( j=0; j<N ; j++ )
        cout << setw( 3 ) << a[i][j] ;
```

```
        cout << endl ;
      }
  }
```

【解答】

```
  1   3   6  10  15
  2   5   9  14   0
  4   8  13   0   0
  7  12   0   0   0
 11   0   0   0   0
```

4.

```
#include <iostream>
using namespace std;
int f(int [],int);
int main()
{ int a[] = { -1, 3, 5, -7, 9, -11 } ;
  cout << f( a, 6 ) << endl ;
}
int f( int a[], int size )
{ int i, t=1 ;
  for( i=0 ; i<size; i ++ )
     if( a[i]>0 )  t *= a[i] ;
  return t;
}
```

【解答】

```
135
```

5.

```
#include <iostream>
using namespace std;
int f( int [][3], int, int ) ;
int main()
{ int a[][3] = { 0, 1, 2, 3, 4, 5, 6, 7, 8 } ;
  cout << f( a, 3, 3 ) << endl ;
}
int f( int a[][3], int row, int col )
{ int i, j, t=1 ;
  for( i=0; i<row; i ++ )
     for( j=0; j<col; j++ )
     { a[i][j] ++ ;
       if( i == j )  t *= a[i][j] ;
     }
  return t ;
}
```

【解答】

```
45
```

6.

```
#include <iostream>
using namespace std;
void test1( int *a1 )
{ a1 = new int( 5 ) ;
  cout << "*a1 = " << *a1 << endl ;
}
void test2(int * & a2)
{ a2 = new int( 5 ) ;
  cout << "*a2 = " << *a2 << endl ;
}
int main()
{ int *p = new int( 1 ) ;
  test1( p ) ;
  cout << "test1: *p1 = " << *p << endl ;
  test2( p ) ;
  cout << "test2: *p2 = " << *p << endl ;
}
```

【解答】

```
*a1= 5
test1: *p1= 1
*a2= 5
test2: *p2= 5
```

7.

```
#include <iostream>
using namespace std;
int main()
{ char s[] = "abccda" ;
  int i ; char c ;
  for( i = 1 ; ( c=s[i] ) != '\0'; i ++ )
    { switch( c )
       { case 'a' : cout << '%' ; continue ;
         case 'b' : cout << '$' ; break ;
         case 'c' : cout << '*' ; break ;
         case 'd' : continue ;
       }
      cout << '#' << endl ;
    }
}
```

【解答】

```
$#
*#
*#
%
```

8.

```
#include <iostream>
using namespace std;
int main()
{ char *str[] = { "c++", "basic", "pascal" } ;
  char **p ;
  int i ;
  p = str ;
  for( i=0 ; i<3 ; i++ )
      cout << *( p+i ) << endl ;
}
```

【解答】

c++

basic

pascal

9.

```
#include <iostream>
using namespace std;
int main()
{ char s1[] = "Fortran" , s2[] = "Foxpro" ;
  char *p , *q ;
  p = s1 ;  q = s2 ;
  while( *p && *q )
   { if ( *p == *q )
     cout << *p ;
     p ++ ;
     q ++ ;
   }
  cout << endl ;
}
```

【解答】

For

10.

```
#include <cstring>
#include <iostream>
using namespace std;
int main()
{ char str[][10] = { "vb", "pascal", "c++" }, s[10] ;
  strcpy_s( s , ( strcmp( str[0], str[1] ) < 0 ? str[0] : str[1] ) ) ;
  if( strcmp( str[2], s ) < 0 )
     strcpy_s( s, str[2] ) ;
  cout << s << endl ;
}
```

【解答】

C++

4.3 思考题

1．数组说明语句要向编译器提供什么信息？请写出一维数组、二维数组说明语句的形式。

【解答】

数组说明语句要向编译器提供数组名（标识符），数组元素的类型、数组维数、数组长度（元素的个数）等信息。

一维数组说明语句为： 类型 数组名[表达式]

二维数组说明语句为： 类型 数组名[表达式$_1$][表达式$_2$]

2．数组名、数组元素的区别是什么？归纳一维数组元素地址、元素值不同的表示形式。若有说明

```
int aa [3], *pa=aa;
```

请使用 aa 或 pa，写出 3 个以上与 aa[2] 等价的表达式。

【解答】

数组名是一个标识符，执行代码中代表数组的地址，即指向数组起始位置的指针；而数组元素是下标变量，性质相当于普通变量。

对一维数组 aa 第 i 个元素的地址可以表示为： &aa[i] aa+i;

对一维数组 aa 第 i 个元素的值可以表示为： a[i] *(a+i);

与 aa[2]等价的表达式：

*(aa+2) *(&a[2]) *(pa+2) pa[2]

3．要把一维数组 int a[m*n] 的元素传送到二维数组 int b[m][n] 中，即在程序中要执行

```
b[i][j]=a[k];
```

请写出 k→i,j 的下标变换公式，并用程序验证。

【解答】

转换公式： i=k/n j=k%n

验证程序：

```
#include <iostream>
using namespace std;
int main()
{ const int M=3,N=4;
  int k,a[M*N]={1,2,3,4,5,6,7,8,9,10,11,12},b[M][N];
  int i,j;
  cout<<"array a:"<<endl;
  for( k=0; k<M*N; k++ )
     b[k/N][k%N] = a[k];
  for( k=0; k<M*N; k++ )
    cout<<a[k]<<'\t';
  cout<<endl;
  cout<<"**After convert**"<<endl;
  cout<<"array b:"<<endl;
  for(i=0;i<M;i++)
    { for(j=0;j<N;j++) cout<<b[i][j]<<'\t';
      cout<<endl;
    }
}
```

4．有以下函数

```
void query()
{ int *p;
  p=new int[3];
    //……
  delete []p;
  p=new double[5];
    //……
  delete []p;
}
```

出现了编译错误。请分析错误的原因，并把上述程序补充完整，上机验证你的判断。

【解答】

在语句 p=new double[5]; 中企图把动态浮点型数组的地址写入整型指针 p，造成错误。错误为 error C2440: “=”: 无法从“double *”转换为“int *”。

改正方法：增加一个 double*q 指针。

```
void query()
{ int *p;
  p=new int[3];
  delete [] p;
  //.....
  double *q;
  q=new double[5];
  //.....
  delete []q;
}
```

5. 有以下程序根据输入值，设计功能是调用函数 create 建立并初始化动态数组，令 a[i]=i。但该程序运行后不能得到期望结果，请分析程序的错误原因并修改之。

```
#include <iostream>
using namespace std;
void create(int *, int);
int main()
{ int *a = NULL, len;
  cin>>len;
  create(a,len);
  for( int i = 0; i<len; i++ )
     cout << a[i] << "   ";
  cout << endl;
  delete []a;
  a = NULL ;
}
void create(int *ap, int n)
{ ap=new int[n];
  for(int i=0; i<n; i++) ap[i]=i;
}
```

【解答】

函数create中，指针参数int*ap是传地址值的参数。调用函数时接受实际参数a的值（地址值）作为初始值。ap仅是局部变量，ap=new int[n]获得新的地址值，函数执行完毕返回，ap被释放，完全与实际参数a无关。程序没有编译错误，但main不能获得动态数组。修改方法是把ap改为指针引用参数。

```
void create(int *&,int);              //函数原型声明,使用引用参数
void create(int *&ap,int n)           //函数定义
{ ap=new int[n];
  for(int i=0;i<n;i++)
     ap[i]=i;
}
```

4.4 编程题

1．已知求成绩的平均值和均方差公式：$ave=\frac{\sum_{i=1}^{n}s_i}{n}$，$dev=\sqrt{\frac{\sum_{i=1}^{n}(s_i-ave)^2}{n}}$，其中 n 为学生人数，$s_i$ 为第 i 个学生成绩。求某班学生的平均成绩和均方差。

【解答】

```
#include<iostream>
#include<cmath>
using namespace std;
void aveMsd( double [], int, double &, double & );  //求平均值和均方差值函数
int main()
{ double s[] = { 76, 85, 54, 77, 93, 83, 90, 67, 81, 65 };
  double ave, msd;
  int i,n;
  n = sizeof( s )/sizeof( double );     //求数组元素的个数
  cout<<"学生成绩：";
  for( i=0; i<n; i++ )
   cout<<s[i]<<"  ";
  cout<<endl;
  aveMsd( s, n, ave, msd );
  cout << "平均值：" << ave << endl << "均方差值：" << msd << endl;
}
void aveMsd( double s[], int n, double &ave, double &msd )
{ int i;
  double sum1=0, sum2=0;
  for( i=0; i<n; i++ )       //求平均值
    sum1 += s[i];
  ave = sum1/n;
  for( i=0; i<n; i++ )       //求均方差
    sum2 += pow( s[i]-ave, 2 );
  msd = sqrt( sum2/n );
}
```

2．用随机函数产生 10 个互不相同的两位整数存放到一维数组中，并输出其中的素数。

【解答】

```
#include<iostream>
#include<cmath>
#include <cstdlib>
#include<ctime>
using namespace std;
int main()
{ int a[10],i,j;
  srand( int( time(0)) );        //为随机数生成器设置种子值
  for( i=0; i<10; i++ )
  { 1: a[i] = rand();            //产生随机数存放到数组中
    if ( a[i]<10 || a[i]>=100 )        //获取指定范围数据
       goto 1;
    for( j=0; j<i; j++ )               //排除相同数据
       if( a[i]==a[j] )
         goto 1;
  }
for( i=0; i<10; i++ )
   cout << a[i] << "   ";
cout << endl;
for( i=0; i<10; i++ )
  { double m=sqrt( double (a[i]) );
    for( j=2; j<=m; j++)
       if( a[i] % j == 0 )break;
      if( j>m )
         cout << a[i] << "   ";
    }
  cout << "是素数!" << endl;
}
```

3. 将一组数据从大到小排列后输出，要求显示每个元素及它们在原数组中的下标。

【解答】

```
#include<iostream>
using namespace std;
int main()
{ int a[] = { 38, 6, 29, 1, 25, 20, 6, 32, 78, 10 };
  int index[10];       //记录下标的数组
  int i,j,temp;
for( i=0; i<10; i++ )
   index[i] = i;
for( i=0; i<=8; i++ )
   for( j=i+1; j<=9; j++ )
     if( a[i] < a[j] )
       { temp = a[i]; a[i] = a[j]; a[j] = temp;
         temp = index[i]; index[i] = index[j]; index[j] = temp;
       }
```

```
    for( i=0; i<10; i++ )
       cout << a[i] << '\t' << index[i] << endl;
  }
```

4. 从键盘上输入一个正整数，判别它是否为回文数。所谓回文数是指正读和反读都一样的数。例如，123321 是回文数。

【解答】

```
#include<iostream>
using namespace std;
int main()
{ int b[10], i, j, k, flag ;
  long num, n ;
  cout << "num=" ;  cin >> num;
  k = 0;
  n = num;
  do                                  //拆分整数，把各数字放入数组 b
  { b[k++] = n % 10;
    n = n/10;
  } while( n != 0);
  flag=1;                             //判断标志
  i=0; j=k-1;                         //设置指示下标的指针
  while(i<j)
    if( b[i++] != b[j--] )            //对称位置元素不相等
       { flag = 0;
         break ;
       }
    if( flag ) cout << num << "是回文数！" << endl;
       else cout << num << "不是回文数！" << endl;
}
```

5. 把两个升序排列的整型数组合并为一个升序数组。设计好你的算法，以得到较高的运行效率。

【解答】

```
#include<iostream>
using namespace std;
void merge(const int a[],int na, const int b[],int nb, int c[],int nc);
int main()
{ int a[4] = { 1, 2, 5, 7 };
  int b[8] = { 3, 4, 8, 8, 9, 10, 11, 12 };
  int c[12];
  int i;
  merge( a,4,b,8,c,12 );
  for( i=0; i<12; i++ )
    cout << c[i] << "   ";
  cout << endl;
}
void merge(const int a[],int na, const int b[],int nb, int c[],int nc)
```

```
{ int i,j,k;
  i = j = k = 0;
  while( i<na && j<nb )
  if( a[i] > b[j] )                    //当a[i]>b[j]，把b[i]写入数组c
    { c[k] = b[j]; k++; j++; }
  else                                 //当a[i]<=b[j]，把a[i]写入数组c
    { c[k] = a[i]; k++; i++; }
  while( i<na )
    { c[k] = a[i]; i++; k++; }      //把数组a的剩余元素写入数组c
    while( j<nb )
      { c[k] = b[j]; k++; j++; }      //把数组b的剩余元素写入数组c
}
```

6. 输入一个表示星期几的数，然后输出相应的英文单词。要求使用指针数组实现。

【解答】

```
#include<iostream>
using namespace std;
int main()
{ char *weekday[7] = { "sunday", "monday", "tuesday",
                       "wednesday", "thursday", "friday", "saturday" };
  int d;
  cout << "please input week day: ";
  cin >> d;
    if( d>=0 && d<=6 )cout << d << "---" << *( weekday + d ) <<endl;
       else cout << "input error!" << endl;
}
```

7. 编写函数：

（1）在一个二维数组中形成如以下形式的 n 阶矩阵：

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 1 & 1 \\ 3 & 2 & 1 & 1 & 1 \\ 4 & 3 & 2 & 1 & 1 \\ 5 & 4 & 3 & 2 & 1 \end{pmatrix}$$

（2）去掉靠边元素，生成新的 n-2 阶矩阵；

（3）求矩阵主对角线下元素之和；

（4）以方阵形式输出数组。

在 main 函数中调用以上函数进行测试。

【解答】

```
#include<iostream>
using namespace std;
void create( int *&app, int n );
void del( int *&app, int *&bpp, int n );
int maindiagonal( int *&app, int n );
void output( int *&app, int );
int main()
```

```
{ int *ap = NULL, *bp = NULL, n;
  cout << "输入矩阵的阶：";
  cin >> n;
  create( ap,n );
  cout << "\n 形成矩阵：\n";
  output( ap, n );
  cout << "去掉靠边元素生成的矩阵：\n";
  del( ap,bp,n );
  output( bp,n-2 );
  cout << "主对角线元素之和：" << maindiagonal( ap, n ) <<endl;
}
//形成 n 阶矩阵函数
void create( int * &app, int n )
{ app = new int[ n*n ];
  int i,j,k = 0;
  for( i=0; i<n; i++ )
    for( j=0; j<n; j++ )
     {  if( i<=j ) app[k] = 1;
        else app[k] = i-j+1;
        k++;
     }
}
//去掉靠边元素生成 n-2 阶矩阵函数
void del( int *&app, int *&bpp, int n )
{ int i,j,k = 0;
  bpp = new int[ ( n-2 )*( n-2 ) ];
  for ( i=0; i<n; i++ )
     { for( j=0; j<n; j++ )
          if ( i && j && i<n-1 && j <n-1 )
             { bpp[k]= *( app + i*n + j );
               k++;
             }
     }
}
//求主对角线元素之和函数
int maindiagonal( int *&app, int n )
{ int i,j,k = 0,sum = 0;
  for ( i=0; i<n; i++ )
    { for( j=0; j<n; j++ )
        if( i==j )
          sum += *( app + i*n + j);
     }
  return sum;
}
```

```
//以方阵的形式输出数组函数
void output( int *&app, int n )
{ int i,j;
  for ( i=0; i<n ; i++ )
     { for( j=0; j<n; j++ )
          cout << *( app + i*n + j) <<  '\t';
       cout<<endl;
     }
    cout<<endl;
}
```

8．设某一城市三个百货公司某个季度销售电视机的情况和价格如下表格所示。编写程序，将每个表数据以数组存放，求各个百货公司的电视机营业额。

牌　号 / 公司	康佳	TCL	长虹
第一百货公司	300	250	150
第二百货公司	200	240	200
第三百货公司	280	210	180

牌号	价格
康佳	3500
TCL	3300
长虹	3800

【解答】

```
#include<iostream>
using namespace std;
int main()
{ long s[][3] = { { 300, 250, 150 }, { 200, 240, 200},{ 280, 210, 180 } };
  long p[] = { 3500, 3300, 3800 };
  int i,j;
  double sum;
  for( i=0; i<3; i++ )
  { sum=0;
    for( j=0; j<3; j++)
      sum += s[i][j] * p[j];
    cout << "第" << i+1 << "百货公司的电视机营业额:  " << sum << endl;
  }
}
```

9．设计函数求一整型数组的最小元素及其下标。在主函数中定义和初始化该整型数组，调用该函数，并显示最小元素值和下标值。

【解答】

```
#include<iostream>
using namespace std;
int fmin(int [], int);
int main()
{ int a[ ] = { 73, 85, 62, 95, 77, 56, 81, 66, 90, 80 };
  int index;
  index = fmin( a, sizeof(a)/sizeof(int) );
  cout << "The minnum number is : " << a[index] << endl;
```

```
  cout << "The index is : " << index << endl;
}
int fmin( int a[], int size )
{ int i,min = a[0], index = 0;
  for( i=0; i<size; i++ )
   if( a[i]<min )
      {  min = a[i]; index = i; };
  return index;
}
```

10. 假设有从小到大排列的一组数据存放在一个数组中，在主函数中从键盘输入一个在该数组的最小值和最大值之间的数，并调用一函数把输入的数插入到原有的数组中，保持从小到大的顺序，并把最大数挤出。要求在主函数中输出改变后的数组。

【解答】

```
#include<iostream>
using namespace std;
void insert( int a[],int,int );
int main()
{ int a[] = { 10, 12, 23, 25, 48, 48, 53, 58, 60, 78 };
  int x,n,i;
  cout << "please input insert data: ";
  cin >> x;
  n = sizeof(a)/sizeof(int);          //求数组长度
  insert( a, n, x );                  //插入元素
  for( i=0; i<n; i++ )
    cout << a[i] << "   ";
  cout << endl;
}
void insert( int a[],int n,int x )
{ int i,p,j;
  if ( x<a[n-1] )
   { for( i=1; i<n; i++ )          //查找插入位置
      if( x<a[i] )
       {  p=i; break;
       }
     for( j=n-1; j>=p; j-- )       //后移元素，挤出最大值
       a[j] = a[j-1];
       a[p] = x;                   //插入元素
    }
}
```

11. 编写程序，按照指定长度生成动态数组，用随机数对数组元素赋值，然后逆置该数组元素。例如，对数组 A[5]，初值为{6，3，7，8，2}，逆置后的值是{2，8，7，3，6}。程序输出逆置前、后的数组元素序列。

【解答】

```
#include<iostream>
```

```
#include <cstdlib>
#include<ctime>
using namespace std;
void printarray(int *p,int n);
void adverse(int *p,int n);
int main()
{ int *p,n,i;
  cout<<"请输入数组长度:";
  cin>>n;
  p=new int [n];              // 产生动态数组
  srand(time(0));
  for(i=0;i<n;i++)            //产生随机数并存放到动态数组中
    *(p+i)=rand()/1000;
  cout<<"动态数组:";
  printarray(p,n);              // 输出动态数组
  adverse(p,n);                 // 对数组逆置
  cout<<"逆置数组:";
  printarray(p,n);              // 输出逆置数组
}
 // 输出数组函数
void printarray(int *p,int n)
{ int i;
   for( i=0; i<n; i++ )
     { if (i % 5==0) cout<<endl;   // 控制一行输出 5 个数据
         cout<<"array["<<i<<"]="<<*(p+i)<<" ";
     }
   cout<<endl;
 }
// 对数组逆置函数
void adverse(int *p,int n)
{ int i,t;
  for (i=0;i<n/2;i++)
    { t=*(p+i);
      *(p+i)=*(p+n-i-1);
      *(p+n-i-1)=t;
    }
```

12. 把一个字符串插入到另一个字符串中指定的位置。

【解答】

```
#include<iostream>
using namespace std;
int main()
{ int p, i, j, k;
  char s1[40], s2[40], s3[80];
```

```
cout << "s1=";
cin >> s1;
cout << "s2=";
cin >> s2;
cout << "input insert position:";
cin >> p;                          //输入插入位置
for( i=0; i<p; i++ )
  s3[i] = s1[i];
for( j=0; s2[j] != '\0'; j++ )
    s3[i+j] = s2[j];
  for( k=p; s1[k] != '\0'; k++ )
    s3[j+k] = s1[k];
  s3[j+k] = '\0';
  cout << "s3=" << s3 << endl;
}
```

13. 把某班学生的姓名和学号分别存放到两个数组中，从键盘上输入某一学生学号，查找该学生是否在该班，若找到该学生，则显示出相应的姓名。

【解答】

```
#include<iostream>
using namespace std;
int main()
{ char name[5][20] = { "li ming", "zhang qing",
                       "liu xiao ping", "wang ying", "lu pei" };
  long num[5] = { 20030001, 20030002, 20030005, 20030007, 20030010 };
  int i;
  long snumber;
  cout << "please input studentnumber:";
  cin >> snumber;
  for( i=0; i<5; i++ )
   { if( num[i] == snumber )
      { cout << "Found! The name is :" << name[i] << endl;
        break;
      }
   }
  if( i==5 ) cout << "Can\'t found!" << endl;
}
```

14. 将一组C++关键字存放到一个二维数组中，并找出这些关键字的最小者。

【解答】

```
#include<iostream>
#include <cstring>
using namespace std;
int main()
{ char string[10];
  char str[][10]={ "while", "break", "if", "extern", "void", "auto",
```

```
                    "long", "static", "do", "const" };
    int i;
    strcpy( string, str[0] );
    for( i=0; i<10; i++ )
      if( strcmp(str[i],string)<0 ) strcpy( string, str[i] );
    cout << "The minimum string is:" << string << endl;
  }
```

15. 使用指针函数编写程序，把两个字符串连接起来。

【解答】

```
#include<iostream>
using namespace std;
char *strcat( char *str1,char *str2 )
{ char *p = str1;
  while( *p != '\0' ) p++;
  *p = *str2;
  do { p++;
       str2++;
       *p = *str2;
     } while( *str2 != '\0' );
  return( str1 );
}
int main()
  { char str1[80], str2[80];
    cout << "input str1:";
    cin >> str1;
    cout << "input str2:";
    cin >> str2;
    cout << "str1+str2=" << strcat( str1, str2 ) << endl;
  }
```

16. 使用 string 类，写一个简单文本编辑程序，能够实现基本的插入、删除、查找、替换等功能。

【解答】

略。

习题 5 及其解答

5.1　选择题

1. 有说明

```
struct point
{ int x; int y; }p;
```

正确的赋值语句是（ c ）。

(a) point.x = 1; point.y = 2;　　(b) point={ 1, 2 };

(c) p.x = 1; p.y = 2;　　(d) p = { 1, 2 };

2. 已知有职工情况结构变量emp定义为：

```
struct Date
{ int year;
  int month;
  int day;
};
strnct Employee
{ char name[20] ;
  long  code ;
  Date birth
};
Employee emp ;
```

下列对emp的birth的正确赋值方法是（ d ）。

(a) year=1980; month=5; day=1;

(b) birth.year=1980; birth.month=5; birth.day=1;

(c) emp.year=1980; emp.month=5; emp.day=1;

(d) emp.birth.year=1980; emp.birth.month=5; emp.birth.day=1;

3. 假定有以下说明语句，则下面引用形式错误的是（ b ）。

```
struct Student
{ int num ;
  double score ;
};
Student stu[3]={{1001,80}, {1002,75}, {1003,91}}, *p=stu ;
```

(a) p->num　　(b) (p++).num　　(c) (p++)->num　　(d) (*p).num

4. 若有以下说明语句，则下列错误的引用是（ d ）。

```
struct  Worker
{ int  no;
  char name[20];
```

```
};
Worker w, *p = &w ;
```

(a) w.no　　(b) p->no　　(c) (*p).no　　(d) *p.no

5. s1和s2是两个结构类型变量，若要赋值s1=s2合法，则它们的说明应该是（ c ）。

(a) s1只能接受相同类型的数据成员　　(b) 结构中的成员相同

(c) 同一结构类型的变量　　(d) 存储字节长度一样的变量

5.2 阅读下列程序，写出执行结果。

1.

```
#include <iostream>
using namespace std;
struct Data
 { int n ;
   double score ;
 } ;
int main()
{ Data a[3] = { 1001,87,1002,72,1003,90 } , *p = a ;
  cout << (p++)->n << endl ;
  cout << (p++)->n << endl ;
  cout << p->n++ << endl ;
  cout << (*p).n++ << endl ;
}
```

【解答】

```
1001
1002
1003
1004
```

2.

```
#include <iostream>
using namespace std;
struct Employee
 { char name[ 20 ] ;
   char sex ;
 } ;
void fun( Employee *p )
{ if( (*p).sex == 'm' )
    cout << (*p).name << endl ;
}
int main()
{ Employee emp[5] = { "Liming", 'm', "Wangxiaoping", 'f', "Luwei", 'm' } ;
  int i ;
  for( i=0; i<3; i++ )
    fun( emp+i ) ;
}
```

【解答】

Liming

Luwei

3.

```
#include <iostream>
using namespace std;
struct Node
 { char * s ;
   Node * q ;
 } ;
int main()
{ Node a[ ] = { { "Mary", a+1 }, { "Jack", a+2 }, { "Jim", a } } ;
  Node *p = a ;
  cout << p->s << endl ;
  cout << p->q->s << endl ;
  cout << p->q->q->s << endl ;
  cout << p->q->q->q->s << endl ;
}
```

【解答】

Mary

Jack

Jim

Mary

5.3 思考题

1. 分析以下说明结构的语句

```
struct Node
{ int data;
  Node error;          //错误
  Node * ok;           //正确
};
```

error 和 ok 分别属于什么数据类型？有什么存储要求？error 出错的原因是什么？

【解答】

error 是 Node 结构类型数据成员，错误。原因是结构定义的数据成员若为本身的结构类型，是一种无穷递归。ok 是指向 Node 类型的指针，定义正确，占 4 字节。

2. 本章例 5-5 中用辅助数组对结构数组做关键字排序，有定义

```
person *index[100];
```

index 数组存放结构数组元素的地址。如果把 index 定义改为

```
int index[100];
```

用于存放结构数组元素的下标。可以实现对结构数组的索引排序吗？如何修改程序？请你试一试。

【解答】

可以。关键是通过整型索引数组元素作为下标访问结构数组。表示为：

```
all[pi[i]].name   all[pi[i]].id      all[pi[i]].salary
```

有关程序如下：

```
#include<iostream>
using namespace std;
```

```
struct person           //说明结构类型
{ char name[10];
  unsigned int id;
  double salary;
} ;
void Input( person[], const int );
void Sort( person[], int[],const int );
void Output( const person[], int[],const int );
int main()
{ person allone[100] ;      //说明结构数组
  int index[100];           //说明索引数组
  int total ;
  for(int i=0; i<100; i++) //索引数组元素值初始化为结构数组元素下标
     index[i]=i ;
  cout<<"输入职工人数：";
  cin>>total;
  cout<<"输入职工信息：\n";
  Input(allone,total);
  cout<<"以工资做关键字排序\n";
  Sort(allone,index, total);
  cout<<"输出排序后信息：\n";
  Output(allone,index,total);
}
void Input( person all[], const int n )
{ int i ;
  for( i=0; i<n; i++ )      // 输入数据
   { cout<<i<<": 姓名: ";
     cin>>all[i].name;
     cout<<"编号: ";
     cin >> all[i].id;
     cout<<"工资: ";
     cin >> all[i].salary ;
   }
}
void Sort(person all[], int pi[], const int n)
{ int i,j;
int t;                      //交换用中间变量
for(i=1; i<n; i++)          //以成员salary做关键字排序
{ for(j=0; j<=n-1-i; j++)
   if(all[pi[j]].salary>all[pi[j+1]].salary)     //通过索引数组访问结构数组元素
        { t=pi[j];                               //交换索引数组元素值
          pi[j]=pi[j+1];
          pi[j+1]= t;
        }
```

```
    }
  }
  void Output(const person all[], int pi[], const int n)
  { for( int i=0; i<n; i++ )       // 输出排序后数据
      cout<<all[pi[i]].name<<'\t'<<all[pi[i]].id<<'\t'<<all[pi[i]].salary<<endl;
  }
```

3. 有以下结构说明和遍历单向链表的函数。函数内有错误吗？是什么性质的错误？请上机验证你的分析。

```
struct Node
 { int data;  Node * next; };
void ShowList( Node *head )
  { while( head )
    { cout << head->date << '\n' ;
      head ++ ;
    }
  }
```

【解答】

head++错误，原因是动态链表的结点存放不是连续顺序的内存空间，它们是逐个结点通过new建立的，所以不能用++做地址偏移运算。应该用：

```
head=head->next
```

5.4 编程题

1. 使用结构类型表示复数。设计程序输入两个复数，可以选择进行复数的+、－、×或÷运算，并输出结果。

【解答】

```
#include <iostream>
#include <iomanip>
using namespace std;
struct complex
 {  double re,im;
 };
int main()
{ complex a,b,c; char oper;
  cout << "输入复数 a 的实部和虚部: ";
  cin >> a.re >> a.im;
  cout << "输入复数 b 的实部和虚部:";
  cin >> b.re >> b.im;
  cout << "输入运算符: ";
  cin >> oper;
  switch ( oper )
  {  case '+': c.re=a.re+b.re; c.im=a.im+b.im; break;
     case '-': c.re=a.re-b.re; c.im=a.im-b.im; break;
     case '*': c.re=a.re*b.re-a.im*b.im;
               c.im=a.im*b.re+a.re*b.im; break;
```

```
    case '/': c.re=(a.re*b.re+a.im*b.im)/(b.re*b.re+b.im*b.im);
              c.im=(a.im*b.re-a.re*b.im)/(b.re*b.re+b.im*b.im);
              break;
    default: cout << "input error!" << endl;
    return 0;
 }
cout << "c=" << c.re;
cout << setiosflags( ios::showpos );
cout << c.im << "i" << endl;
    return 0;
}
```

2．把一个班的学生姓名和成绩存放到一个结构数组中，寻找和输出最高分者。

【解答】

```
#include <iostream>
using namespace std;
int main()
{ struct data
    { char name[12];
      double score;
    }a[ ] = {"李小平",90,"何文章",66,"刘大安",87,"汪立新",93,"罗建国",
                          78,"陆丰收",81,"杨勇",85,"吴一兵",55,"伍晓笑",68,"张虹虹",93};
  double max = a[0].score;
  int i,n = sizeof(a) / sizeof(data);
  for( i=1; i<n; i++ )
     if( a[i].score > max ) max = a[i].score;
  for( i=0; i<n; i++ )
     if( a[i].score == max ) cout << a[i].name << endl;
}
```

3．使用结构表示 X—Y 平面直角坐标系上的点，编写程序顺序读入一个四边形的四个顶点坐标，判别由这四个顶点的连线构成的图形是否为正方形、矩形或其它四边形。要求定义求两个点距离的函数使用结构参数。

【解答】

```
#include <iostream>
#include <cmath>
using namespace std;
struct point
  { double x; double y;
  };
double d( point p1, point p2 )
 {  return sqrt( pow( p1.x-p2.x,2 )+pow( p1.y-p2.y,2 ) );
 }
int main()
{ int i;  point p[5];
  for( i=1; i<=4; i++ )
```

```
  { cout << "输入第" << i << "个顶点的横坐标和纵坐标: ";
    cin >> p[i].x >> p[i].y;
  }
if( fabs( d( p[1],p[2] ) - d( p[3],p[4] ))<=1e-8
      && fabs( d( p[1],p[4] ) - d( p[2],p[3] ))<=1e-8
      && fabs( d( p[1],p[3] ) - d( p[2],p[4] ))<=1e-8)
   if( fabs( d( p[1],p[2] ) - d( p[2],p[3] ))<1e-8 )
       cout << "四个顶点构成的图形为正方形！" << endl;
    else cout << "四个顶点构成的图形为矩形！" << endl;
     else cout << "四个顶点构成的图形为其它四边形！" << endl;
  }
```

4．建立一个结点包括职工的编号、年龄和性别的单向链表，分别定义函数完成以下功能：

（1）遍历该链表输出全部职工信息；

（2）分别统计出男女性职工的人数；

（3）在链表尾部插入新职工结点；

（4）删除指定编号的职工结点；

（5）删除年龄在 60 岁以上的男性职工或 55 岁以上的女性职工结点，并保存在另一个链表中。

用主函数建立简单菜单选择，测试你的程序。

【解答】

```
#include <iostream>
using namespace std;
struct employee
{ int num;
  int age;
  char sex;
  employee *next;
};
employee *head, *head1;
//建立单向链表
employee *create()
{ employee *head, *p, *pend;
  char ch;
  head = NULL;
  cout << "\t 输入数据?(y/n)"; cin >> ch;
  if( ch == 'y' )
    { p = new employee;
      cout << "\t 编号:";  cin >> p->num;
      cout << "\t 年龄:";  cin >> p->age;
      cout << "\t 性别:";  cin >> p->sex;
   }
 else
    goto L0;
while( ch == 'y' )
  {  if( head == NULL ) head = p;
```

```
  else pend->next = p;
pend = p;
cout << "\t 输入数据?(y/n)"; cin>>ch;
if( ch == 'y' )
  { p = new employee;
    cout << "\t 编号:"; cin >> p->num;
    cout << "\t 年龄:"; cin >> p->age;
    cout << "\t 性别:"; cin >> p->sex;
  }
    }
   pend->next = NULL;
   L0: ;
   return head;
}
//显示单向链表中全部职工信息
void show( employee *head )
{ employee *p = head;
  if( !head ) { cout << "\t 空链表!" << endl; goto L1; }
     cout << "\t 链表中的数据是: \n";
  while( p )
  { cout << '\t' << p->num << "," << p->age << "," << p->sex << endl;
     p = p->next;
  }
L1: ;
}
//统计男女职工人数
void count( employee *head )
{ employee *p = head;
  int m, f;
  m = 0; f = 0;
  while( p )
   { if( p->sex == 'm' ) m++;
       else f++;
     p = p->next;
   }
  cout << "\t 男职工人数:" << m << endl;
  cout << "\t 女职工人数:" << f << endl;
}
//在链表尾部插入新结点
 employee *insert()
{ employee *pend = head, *p;
  //在空链表尾部插入新结点
  if( !head )
  { p = new employee;
```

```
  cout << "\t 编号:";  cin >> p->num;
  cout << "\t 年龄:";  cin >> p->age;
  cout << "\t 性别:";  cin >> p->sex;
  head = p;
  p->next = NULL;
  return head;
}
//在链表尾部插入新结点
while( pend->next != NULL )
   { pend = pend->next;
   }
  p = new employee;
  cout << "\t 编号:";  cin >> p->num;
  cout << "\t 年龄:";  cin >> p->age;
  cout << "\t 性别:";  cin >> p->sex;
  pend->next = p;
  pend = p;
  pend->next = NULL;
  return head;
}
//删除指定编号的结点
employee *del( int bh )
{ employee *p, *q;
  if ( !head )
  { cout << "\t 空链表!" << endl;
    goto L2;
  }
  //删除链首结点
  if( head->num == bh )
   { p = head;
     head = head->next;
      delete p;
      cout << "\t 结点已被删除!" << endl;
      goto L2;
   }
  //删除非链首结点
  q = head;
  while( q->next != NULL )
   { if ( q->next->num == bh )
      { p = q->next;       //待删除结点
        q->next = p->next;
        delete p;
       cout << "\t 结点已被删除!" << endl;
        goto L2;
```

```
    }
     q = q->next;
  }
  cout << "\t 找不到需删除结点！" << endl;
  L2: ;
  return ( head );
}
//删除指定年龄段的结点，并把被删除结点保存在另一链表中
employee *delcreate()
{ employee *p, *pd, *pl, *q;
  int flag;
 //建立新链表
  if ( head == NULL )
   { cout << "\t 空链表！" << endl;
     goto L3;
   }
  head1 = NULL;
  pd = new employee;
  p = head;
  flag = 0;
  while ( p != NULL )
   { if( p->age >= 55 && p->age <=60 )
       { pd->num = p->num;
         pd->age = p->age;
         pd->sex = p->sex;
         if( head1 == NULL ) head1 = pd;
           else pl->next = pd;
         pl = pd;
         pd = new employee;
         flag = 1;
       }
     p = p->next;
   }
if ( flag == 0 )
  { cout << "\t 没有需删除的结点！" << endl; goto L3; }
pl->next = NULL;
//显示新链表
cout <<"\t 新链表中的数据是: \n";
p = head1;
while( p )
 { cout << '\t' << p->num << "," << p->age << "," << p->sex << endl;
   p = p->next;
 }
//删除指定年龄的结点
```

```
p = head;
q = p;
while ( p != NULL )
  { if( p->age >= 55 && p->age <= 60)
      if( head->age == p->age )
       { pd = head;                    //待删除结点
         head = head->next;
         delete pd;
         p = head;
       continue;
 }
else
   if( p->next == NULL )
     { pd = p;                         //待删除结点
       q->next = NULL;
      delete pd;
       goto L3;
    }
    else
      { pd = p;                        //待删除结点
        q->next = p->next;
         delete pd;
                p = q->next;
                continue;
              }
    q = p;
    p = p->next;
  }
  L3: return ( head );
}

int main()
{ int choice, bh ;
  L:
   cout << "\n\t\t 请键入操作选择\n" << endl;
   cout << "\t 1 --- 建立单向链表" << endl;
   cout << "\t 2 --- 显示单向链表中全部职工信息" << endl;
   cout << "\t 3 --- 统计男女职工人数" << endl;
   cout << "\t 4 --- 在职工尾部插入新结点" << endl;
   cout << "\t 5 --- 删除指定编号的结点" << endl;
   cout << "\t 6 --- 删除指定年龄的结点，并把被删除结点保存在另一链表中" << endl;
   cout << "\t 0 --- 退出" << endl ;
  cout << "\t\t";
   cin >> choice ;
```

```
switch ( choice )
 { case 1 : head = create() ; goto L ;
   case 2 : show( head );  goto L ;
   case 3 : count( head ); goto L;
   case 4 : head = insert(); goto L;
   case 5 : cout << "\t 输入需删除结点编号:";
            cin >> bh;
             head = del( bh ); goto L;
   case 6 : head = delcreate(); goto L;
        case 0 : cout << " \t 退出程序的运行! \n" << endl ; break ;
        default : cout << "\t 输入错误，请重新输入! \n" << endl ; goto L;
       }
}
```

5. 输入一行字符，按输入字符的反序建立一个字符结点的单向链表，并输出该链表中的字符。

【解答】

```
#include <iostream>
using namespace std;
struct node
 { char ch;
   node *next;
 };
void show( node *head );
int main()
{ node *head, *p;
  char c;
  head = NULL;
  while( (c = getchar()) != '\n' )          //输入一行字符
   { p = new node;                          //建立新结点
     p->ch = c;
     p->next = head;                        //插入表头
     head=p;
   }
   show(head);
}
void show( node *head )                      //输出链表
{ node *p = head;
   cout << "链表中的字符是: \n";
   while( p )
    { cout << p->ch;
       p = p->next;
    }
     cout << endl;
}
```

6. 设有说明语句：

```
struct List
 { int data ;  List * next ;
 } ;
List *head;
```

head 是有序单向链表的头指针。请编写函数：

```
void Count( List * head );
```

计算并输出链表数据相同值的结点及个数。例如，若数据序列为：

2 3 3 3 4 5 5 6 6 6 6 7 8 9 9

则输出结果：

data	number
3	3
5	2
6	4
9	2

请用本章例 5-8 的程序生成有序链表，测试你的函数。

【解答】 略