1172: 结构1：长方形

Time Limit: 1 Sec  Memory Limit: 128 MB  
Submit: 37  Solved: 34  
[[Submit](http://221.203.21.203:8000/rewriter/USTL/http/489629629022/OnlineJudge/submitpage.php?id=1172)][[Status](http://221.203.21.203:8000/rewriter/USTL/http/489629629022/OnlineJudge/problemstatus.php?id=1172)][[Web Board](http://221.203.21.203:8000/rewriter/USTL/http/489629629022/OnlineJudge/bbs.php?pid=1172)]

Description

设计一个表示长方形的结构Rectangle。在main函数中输入n（n<100）个长方形，求这个n的长方形的面积平均值。  
要求：  
1、main函数中定义一个结构数组存储n个长方形；  
2、输入长方形使用数组下标表示法  
3、计算长方形面积和使用指针表示法（建议）

Input

输入格式：第一个整数为长方形个数n，后续为n个长方形的长和宽。

Output

Sample Input

3 2.5 3.4 6 10.5 6 9

Sample Output

41.8333

#include<iostream>

using namespace std;

struct F

{ double l,w;};

int main()

{ F a[1000],\*p;

double n,s;

int i;

cin>>n;

for(i=0;i<n;i++)

cin>>a[i].l>>a[i].w;

for(s=0,p=a;(\*p).l;p++)

s+=p->l\*p->w;// s+=(\*p).l\*(\*p).w;两种表达

cout<<s/n;

return 0;

}

1173: 类与对象1：Point类1

Time Limit: 1 Sec  Memory Limit: 128 MB  
Submit: 114  Solved: 105  
[[Submit](http://221.203.21.203:8000/rewriter/USTL/http/489629629022/OnlineJudge/submitpage.php?id=1173)][[Status](http://221.203.21.203:8000/rewriter/USTL/http/489629629022/OnlineJudge/problemstatus.php?id=1173)][[Web Board](http://221.203.21.203:8000/rewriter/USTL/http/489629629022/OnlineJudge/bbs.php?pid=1173)]

Description

定义一个Point类，数据成员包括私有数据成员为double类型的点坐标x，y；成员函数包括构造函数Point（用于实现对数据成员x，y的初始化），成员函数Display（用于输出点坐标x、y，输出格式为点坐标用逗号分隔并半角圆括号括起来）。

main函数如下(不得修改main函数)：

int main()

{

       double x,y;

       cin>>x>>y;

       Point p1(x,y);

       p1.Display();

       return 0;

}

Input

Output

Sample Input

12.5 22.7

Sample Output

(12.5,22.7)

#include<iostream>

using namespace std;

class Point

{

private:

double x,y;

public:

Point (double x,double y);

void Display();

};

Point::Point (double x,double y)

{

this->x=x;

this->y=y;

}

void Point::Display()

{

cout<<"("<<x<<","<<y<<")"<<endl;

}

int main()

{

double x,y;

cin>>x>>y;

Point p1(x,y);

p1.Display();

return 0;

}

1174: 类与对象2：Point类2

Time Limit: 1 Sec  Memory Limit: 128 MB  
Submit: 100  Solved: 78  
[[Submit](http://221.203.21.203:8000/rewriter/USTL/http/489629629022/OnlineJudge/submitpage.php?id=1174)][[Status](http://221.203.21.203:8000/rewriter/USTL/http/489629629022/OnlineJudge/problemstatus.php?id=1174)][[Web Board](http://221.203.21.203:8000/rewriter/USTL/http/489629629022/OnlineJudge/bbs.php?pid=1174)]

Description

定义一个Point类，数据成员包括私有数据成员为double类型的点坐标x，y；成员函数包括构造函数Point（用于实现对数据成员x，y的初始化）、成员函数Set（用于改变数据成员x、y的值）、成员函数Display（用于输出点坐标x、y，输出格式为点坐标用逗号分隔并半角圆括号括起来）。

main函数如下(不得修改main函数)：

int main()

{

double x1,y1,x2,y2;

cin>>x1>>y1;

cin>>x2>>y2;

Point p1(x1,y1);

p1.Display();

p1.Set(x2,y2);

p1.Display();

return 0;

}

Input

Output

Sample Input

10 25.5 5.5 20

Sample Output

(10,25.5)

(5.5,20)

#include<iostream>

using namespace std;

class Point

{

private:

double x,y;

public:

Point (double x=0,double y=0);

void Display();

void Set(double,double);

};

Point::Point (double x,double y)

{

this->x=x;

this->y=y;

}

void Point::Set(double x,double y)

{

this->x=x;

this->y=y;

}

void Point::Display()

{

cout<<"("<<x<<","<<y<<")"<<endl;

}

int main()

{

double x1,y1,x2,y2;

cin>>x1>>y1;

cin>>x2>>y2;

Point p1(x1,y1);

p1.Display();

p1.Set(x2,y2);

p1.Display();

return 0;

}

1175: 类与对象3：Point类3

Time Limit: 1 Sec  Memory Limit: 128 MB  
Submit: 74  Solved: 57  
[[Submit](http://221.203.21.203:8000/rewriter/USTL/http/489629629022/OnlineJudge/submitpage.php?id=1175)][[Status](http://221.203.21.203:8000/rewriter/USTL/http/489629629022/OnlineJudge/problemstatus.php?id=1175)][[Web Board](http://221.203.21.203:8000/rewriter/USTL/http/489629629022/OnlineJudge/bbs.php?pid=1175)]

Description

定义一个Point类，数据成员包括私有数据成员为double类型的点坐标x，y；成员函数包括构造函数Point（用于实现对数据成员x，y的初始化）、成员函数Set（用于改变数据成员x、y的值）、成员函数LeftMove（点坐标向左移动detax）、成员函数上移UpMove（点坐标上移detay）、成员函数Display（用于输出点坐标x、y，输出格式为点坐标用逗号分隔并半角圆括号括起来）。

main函数如下(不得修改main函数)：

int main()

{

double x1,y1,x2,y2,dx,dy;

cin>>x1>>y1;

cin>>dx;

cin>>dy;

cin>>x2>>y2;

Point p1;

p1.Display();

p1.Set(x1,y1);

p1.Display();

p1.LeftMove(dx);

p1.Display();

p1.UpMove(dy);

p1.Display();

Point p2(x2,y2);

p2.Display();

return 0;

}

Input

Output

Sample Input

10 25.5 5 10 5.5 20

Sample Output

(0,0)

(10,25.5)

(5,25.5)

(5,35.5)

(5.5,20)

#include<iostream>

using namespace std;

class Point

{

private:

    double x,y;

public:

     Point (double x=0,double y=0);

   void   Display();

     void Set(double,double);

     void UpMove(double );

     void LeftMove(double );

};

void Point::LeftMove(double dx)

{

   x-=dx;

}

void Point::UpMove(double dy)

{

   y+=dy;

}

Point::Point (double x,double y)

{

  this->x=x;

  this->y=y;

}

void Point::Set(double x,double y)

{

    this->x=x;

   this->y=y;

}

void  Point::Display()

{

   cout<<"("<<x<<","<<y<<")"<<endl;

}

int main()

{

    double x1,y1,x2,y2,dx,dy;

    cin>>x1>>y1;

    cin>>dx;

    cin>>dy;

    cin>>x2>>y2;

    Point p1;

    p1.Display();

    p1.Set(x1,y1);

    p1.Display();

    p1.LeftMove(dx);

    p1.Display();

    p1.UpMove(dy);

    p1.Display();

    Point p2(x2,y2);

    p2.Display();

    return 0;

}

1176: 类与对象4：Rectangle类1

Time Limit: 1 Sec  Memory Limit: 128 MB  
Submit: 61  Solved: 53  
[[Submit](http://221.203.21.203:8000/rewriter/USTL/http/489629629022/OnlineJudge/submitpage.php?id=1176)][[Status](http://221.203.21.203:8000/rewriter/USTL/http/489629629022/OnlineJudge/problemstatus.php?id=1176)][[Web Board](http://221.203.21.203:8000/rewriter/USTL/http/489629629022/OnlineJudge/bbs.php?pid=1176)]

Description

定义一个长方形类Rectangle，私有数据成员为double型width、height（表示长方形的宽和高），成员函数包括构造函数Rectangle（用于实现对数据成员width、height的初始化）、成员函数GetArea（计算并返回长方形的面积）。

main函数如下（不得修改main函数）：

int main()

{

double width,height;

cin>>width>>height;

Rectangle r1;

cout<<r1.GetArea()<<endl;

Rectangle r2(width,height);

cout<<r2.GetArea()<<endl;

return 0;

}

Input

Output

Sample Input

10.2 25.5

Sample Output

0

260.1

#include<iostream>

using namespace std;

class Rectangle

{

private:

double width,height;

public:

Rectangle(double width=0,double height=0);

double GetArea();

};

Rectangle::Rectangle(double width,double height)

{ this->width=width;

this->height=height;

}

double Rectangle::GetArea()

{

return width\*height;

}

int main()

{

double width,height;

cin>>width>>height;

Rectangle r1;

cout<<r1.GetArea()<<endl;

Rectangle r2(width,height);

cout<<r2.GetArea()<<endl;

return 0;

}

1177: 类与对象5：Rectangle类2

Time Limit: 1 Sec  Memory Limit: 128 MB  
Submit: 51  Solved: 48  
[[Submit](http://221.203.21.203:8000/rewriter/USTL/http/489629629022/OnlineJudge/submitpage.php?id=1177)][[Status](http://221.203.21.203:8000/rewriter/USTL/http/489629629022/OnlineJudge/problemstatus.php?id=1177)][[Web Board](http://221.203.21.203:8000/rewriter/USTL/http/489629629022/OnlineJudge/bbs.php?pid=1177)]

Description

定义一个长方形类Rectangle，私有数据成员为double型width、height（表示长方形的宽和高），成员函数包括构造函数Rectangle（用于实现对数据成员width、height的初始化，默认宽和高都为10）、成员函数（GetArea计算并返回长方形的面积）、成员函数Expand（用于实现对数据成员width、heigh的值膨胀整数n倍 ）。

main函数如下（不得修改main函数）：

int main()

{

double width,height;

int n;

cin>>width>>height;

cin>>n;

Rectangle r1;

cout<<r1.GetArea()<<endl;

Rectangle r2(width,height);

cout<<r2.GetArea()<<endl;

r2.Expand(n);

cout<<r2.GetArea()<<endl;

return 0;

}

Input

Output

Sample Input

10.2 25.5 3

Sample Output

100

260.1

2340.9

#include<iostream>

using namespace std;

class Rectangle

{

private:

double width,height;

public:

Rectangle(double width=10,double height=10);

double GetArea();

void Expand(int );

};

Rectangle::Rectangle(double width,double height)

{ this->width=width;

this->height=height;

}

double Rectangle::GetArea()

{

return width\*height;

}

void Rectangle::Expand(int n)

{

this->width=n\*width;

this->height=n\*height;

}

int main()

{

double width,height;

int n;

cin>>width>>height;

cin>>n;

Rectangle r1;

cout<<r1.GetArea()<<endl;

Rectangle r2(width,height);

cout<<r2.GetArea()<<endl;

r2.Expand(n);

cout<<r2.GetArea()<<endl;

return 0;

}

1178: 类与对象6：Rectangle类3

Time Limit: 1 Sec  Memory Limit: 128 MB  
Submit: 46  Solved: 44  
[[Submit](http://221.203.21.203:8000/rewriter/USTL/http/489629629022/OnlineJudge/submitpage.php?id=1178)][[Status](http://221.203.21.203:8000/rewriter/USTL/http/489629629022/OnlineJudge/problemstatus.php?id=1178)][[Web Board](http://221.203.21.203:8000/rewriter/USTL/http/489629629022/OnlineJudge/bbs.php?pid=1178)]

Description

定义一个长方形类Rectangle，私有数据成员为double型width、height表示长方形的宽和高），成员函数包括构造函数Rectangle（用于实现对数据成员width、height的初始化，默认宽和高都为10）、成员函数GetArea（计算并返回长方形的面积）。

main函数中输入两组宽和高，用于实例化两个类对象R1，R2，判断两个长方形R1、R2面积的大小。

程序执行时:

输入1.5 1.1 3.3 2输出1.65<6.6

输入 3 5 3 5输出15=15

输入3.5 2.0 1.8 1.1 则输出为7>1.98

Input

Output

Sample Input

3.5 2.0 1.8 1.1

Sample Output

7>1.98

#include<iostream>

using namespace std;

class Rectangle

{

private:

double width,height;

public:

Rectangle(double width=10,double height=10);

double GetArea();

};

Rectangle::Rectangle(double width,double height)

{ this->width=width;

this->height=height;

}

double Rectangle::GetArea()

{

return width\*height;

}

int main()

{

double width,height;

cin>>width>>height;

Rectangle r1(width,height);

cin>>width>>height;

Rectangle r2(width,height);

if(r1.GetArea()>r2.GetArea()) cout<<r1.GetArea()<<">"<<r2.GetArea();

if(r1.GetArea()==r2.GetArea()) cout<<r1.GetArea()<<"="<<r2.GetArea();

if(r1.GetArea()<r2.GetArea()) cout<<r1.GetArea()<<"<"<<r2.GetArea();

return 0;

}

## 1179: 类与对象7：Triangle类1

Time Limit: 1 Sec  Memory Limit: 128 MB  
Submit: 244  Solved: 220  
[[Submit](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/submitpage.php?id=1179)][[Status](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/problemstatus.php?id=1179)][[Web Board](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/bbs.php?pid=1179)]

Description

定义一个三角形类Triangle，包括三个double型数据成员a、b和c表示三角形的三条边，成员函数包括构造函数、IsTriangle、GetArea。构造函数用于实现对数据成员的初始化；成员函数IsTriangle判断三条边是否构成三角形，成员函数GetArea返回三角形的面积。

main函数中输入两组三角形的边长，用这两组边长实例化两个三角形类对象T1，T2，若T1、T2均能构成三角形，则输出两个三角形面积之和，否则输出failure。

输入：3 4 5 6 8 10 输出：30

输入：3 4 5 6 7 8  输出：26.3332

输入：1 2 0 3 4 5  输出：failure

Input

Output

Sample Input

3 4 5 6 7 8

Sample Output

26.3332

#include<iostream>

using namespace std;

#include<cmath>

class Triangle

{

 private:

     double a,b,c;

 public:

     Triangle(double a=0,double b=0,double c=0);

         double GetArea();

         bool IsTriangle();

};

Triangle::Triangle(double a,double b,double c)

{   this->a=a;

    this->b=b;

    this->c=c;

}

bool Triangle::IsTriangle()

{

  if(a+b>c&&a+c>b&&b+c>a) return true;

  else return false;

}

double Triangle::GetArea()

{   double p=(a+b+c)/2;

   return sqrt(p\*(p-a)\*(p-b)\*(p-c));

}

int main()

{

    double a,b,c;

    cin>>a>>b>>c;

     Triangle T1(a,b,c);

    cin>>a>>b>>c;

    Triangle T2(a,b,c);

    if(T1.IsTriangle()&&T2.IsTriangle()) cout<<T1.GetArea()+T2.GetArea();

    else cout<<"failure";

    return 0;

}

## 1180: 类与对象8：Triangle类2

Time Limit: 1 Sec  Memory Limit: 128 MB  
Submit: 271  Solved: 216  
[[Submit](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/submitpage.php?id=1180)][[Status](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/problemstatus.php?id=1180)][[Web Board](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/bbs.php?pid=1180)]

Description

定义一个三角形类Triangle，包括三个double型数据成员a、b和c表示三角形的三条边，成员函数包括构造函数、IsTriangle、GetPerimeter。构造函数用于实现对数据成员的初始化；成员函数IsTriangle判断三条边是否构成三角形，成员函数GetPerimeter返回三角形的周长。

main函数中输入两组三角形的边长，用这两组边长实例化两个三角形类对象T1，T2，若T1、T2均能构成三角形，则输出两个三角形周长之差，否则输出failure。

输入：3 4 5 6 8 10 输出： -12

输入：3 4 5 6 7 8 输出：-9

输入：1 2 0 3 4 5 输出：failure

Input

Output

Sample Input

3 4 5 6 7 7.5

Sample Output

-8.5

#include<iostream>

using namespace std;

class Triangle

{

 private:

     double a,b,c;

 public:

     Triangle(double a=0,double b=0,double c=0);

         double GetArea();

         bool IsTriangle();

};

Triangle::Triangle(double a,double b,double c)

{   this->a=a;

    this->b=b;

    this->c=c;

}

bool Triangle::IsTriangle()

{

  if(a+b>c&&a+c>b&&b+c>a) return true;

  else return false;

}

double Triangle::GetArea()

{   double p=a+b+c;

   return p;

}

int main()

{

    double a,b,c;

    cin>>a>>b>>c;

     Triangle T1(a,b,c);

    cin>>a>>b>>c;

    Triangle T2(a,b,c);

    if(T1.IsTriangle()&&T2.IsTriangle()) cout<<T1.GetArea()-T2.GetArea();

    else cout<<"failure";

    return 0;

}

## 1181: 类与对象9：Cylinder类

Time Limit: 1 Sec  Memory Limit: 128 MB  
Submit: 267  Solved: 208  
[[Submit](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/submitpage.php?id=1181)][[Status](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/problemstatus.php?id=1181)][[Web Board](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/bbs.php?pid=1181)]

Description

定义一个圆柱类Cylinder，成员数据为底圆半径r和圆柱高h，成员函数为构造函数，GetArea计算圆柱体的表面积，GetVolume计算圆柱体的体积。（注意π值取3.14）

main函数输入圆柱的半径和高，定义一个类对象，计算并输出圆柱的表面积和体积。

Input

Output

Sample Input

1.5 1.5

Sample Output

28.26 10.5975

#include<iostream>

using namespace std;

class Cylinder

{

 private:

     double r,h;

 public:

     Cylinder(double r=0,double h=0);

         double GetArea();

         double GetVolume();

};

Cylinder::Cylinder(double r,double h)

{   this->r=r;

this->h=h;

}

double Cylinder::GetVolume()

{

  return r\*r\*3.14\*h;

}

double Cylinder::GetArea()

{

   return 2\*r\*r\*3.14+2\*r\*3.14\*h;

}

int main()

{

    double r,h;

    cin>>r>>h;

     Cylinder T1(r,h);

    cout<<T1.GetArea()<<" "<<T1.GetVolume();

    return 0;

}

## 1182: 类与对象10：Person类

Time Limit: 1 Sec  Memory Limit: 128 MB  
Submit: 348  Solved: 204  
[[Submit](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/submitpage.php?id=1182)][[Status](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/problemstatus.php?id=1182)][[Web Board](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/bbs.php?pid=1182)]

Description

设计一个Person类，其属性包括姓名name和身份证号id，其中name为指针类型，id为整型，编写构造函数Person，实现数据成员初始化；编写拷贝构造函数；编写Display函数显示数据成员信息；编写析构函数。

main函数中利用Person类构造函数建立类对象p1，再利用拷贝构造函数建立对象p2，打印每个person类对象的信息。

Input

Output

要求：两个对象的数据分行显示

Sample Input

liming 12

Sample Output

liming 12

liming 12

#include<iostream>

#include <cmath>

#include <string.h>

using namespace std;

class Person

{

  private: char \*name;

            int id;

  public:

          Person(char \*name1,int id1);

          Person(const Person &a);

           ~Person();

           void Display();

};

void Person::Display()

{

  cout<<name<<" "<<id<<endl;

}

Person::~Person()

{

 delete name;

}

Person::Person(const Person &a)

{ int len;

  id=a.id;

  len=strlen(a.name);

  name=new char[len+1];

  strcpy(name,a.name);

}

Person::Person(char \*name1,int id1)

{

    int len=strlen(name1);

  name=new char[len+1];

  id=id1;

  strcpy(name,name1);

}

int main()

{   char name[10];

    int id;

    cin>>name>>id;

    Person p1(name,id);

    p1.Display();

    Person p2(p1);

    p2.Display();

    return 0;

}

## 1183: 类与对象11：Complex类1

Time Limit: 1 Sec  Memory Limit: 128 MB  
Submit: 213  Solved: 138  
[[Submit](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/submitpage.php?id=1183)][[Status](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/problemstatus.php?id=1183)][[Web Board](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/bbs.php?pid=1183)]

Description

定义一个复数类Complex，该类对象存放一个复数的实部和虚部。设计带有默认形参值的构造函数，实部虚部的默认值均为0；设计一个能够输出复数的Display成员函数（当虚部为0时仅输出实部；输出样式如：18.5、10+3i、10-3.5i、-5.8+9.5i、-4.5-9.6i等）。

main函数如下(不得修改main函数)：

int main()

{

double r1,i1;

cin>>r1>>i1;

Complex c1(r1,i1);

c1.Display();

return 0;

}

Input

Output

Sample Input

2.5 -3.8

Sample Output

2.5-3.8i

#include <iostream>

using namespace std;

class Complex

{

public:

      Complex(double x1=0,double y1=0);

      void Display();

private:

    double x,y;

};

void Complex::Display()

{   if(y>0)

    cout<<x<<"+"<<y<<"i"<<endl;

    else if(y<0) cout<<x<<y<<"i"<<endl;

    else cout<<x<<endl;

}

Complex::Complex(double x1,double y1)

{

    x=x1;

    y=y1;

}

int main()

{

    double r1,i1;

    cin>>r1>>i1;

    Complex c1(r1,i1);

    c1.Display();

    return 0;

}

## 1185: 类与对象12：Array类1

Time Limit: 1 Sec  Memory Limit: 128 MB  
Submit: 308  Solved: 180  
[[Submit](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/submitpage.php?id=1185)][[Status](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/problemstatus.php?id=1185)][[Web Board](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/bbs.php?pid=1185)]

Description

数组类Array用于存储一组双精度浮点数，其定义如下，给出各个成员函数实现。

const int MaxSize=100; //数组中元素的最大个数

class Array

{

public:

Array (); //初始化length=0

int Length(); //返回数组中元素实际个数

void Insert(int i, double x); //在下标i处插入x

void Display(); //输出数组中实际元素

private:

double data[MaxSize]; //存储元素

int length; //数组中实际元素个数

};

main函数中输入n个(n<=100)双精度浮点数（以0结束），将这n个数按输入次序存储到同一个Array类对象的数据成员data中（利用成员函数Insert实现），并调用Display输出这些数。

Input

Output

Sample Input

35 3 64 7 6443 0

Sample Output

Length:5

Elements:35 3 64 7 6443

#include<iostream>

using namespace std;

class Array

{

public:

      Array ();      //初始化length=0

      int Length();   //返回数组中元素实际个数

      void Insert(int i, double x);  //在下标i处插入x

       void Display();       //输出数组中实际元素

private:

        double data[1000];  //存储元素

         int length;              //数组中实际元素个数

};

Array::Array()

{

   length=0;

}

void Array::Insert(int i,double x)

{  int z;

   for(z=length-1;z>=i;z--)

  data[z+1]=data[z];

   data[i]=x;

   length++;

}

void Array::Display()

{   int z;

    cout<<"Length:"<<length<<endl<<"Elements:";

    for(z=0;z<length;z++)

  cout<<data[z]<<" ";

    cout<<endl;

}

int Array::Length()

{

    return length;

}

int main()

{

 int i=0;

 Array n;double x;

      while(cin>>x)

      { if(x==0)break;

       n.Insert(i,x);

       i++;

      }

      n.Display();

return 0;

}

## 1186: 类与对象13：Array类2

Time Limit: 1 Sec  Memory Limit: 128 MB  
Submit: 210  Solved: 139  
[[Submit](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/submitpage.php?id=1186)][[Status](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/problemstatus.php?id=1186)][[Web Board](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/bbs.php?pid=1186)]

Description

数组类Array用于存储一组双精度浮点数，在现有类定义基础适当增加必要的成员函数，完成下列任务：

main函数中输入n个(n<100)双精度浮点数（以0结束），将这n个数存储到同一个Array类对象中，并调用Display输出这些数。再输入删除的元素x，从对象中删除所有的值为x的元素，再将剩余元素按照从小到大的顺序排序，并输出排序后结果。

已知数组类的定义（其它成员函数请根据需要添加）如下：

const int MaxSize=100; //数组中元素的最大个数

class Array

{

public:

Array (); //初始化length=0

int Length(); //返回数组中有效元素个数

void Insert(int i, double x); //在下标i处插入x

void Display(); //输出数组中有效元素

private:

double data[MaxSize]; //存储元素

int length; //数组中有效元素个数

};

主函数如下（不允许改动主函数中代码）：

int main()

{

int i=0;

double num;

Array a; //顺序表的初始化

while(1)

{

cin>>num;

if(!num) break;

try{

a.Insert(i++,num); //insert

}

catch(char \*ms)

{

cout<<ms<<endl;

}

}

a.Display ();

double x;

cin>>x;

try

{

int n=a.Delete(x); //delete all x

cout<<"Count of deleted elements:"<<n<<endl;

}

catch(char \*ms)

{

cout<<ms<<endl;

}

a.Sort(); //sort

a.Display();

return 0;

}

Input

Output

Sample Input

23 2 2 344 3 32 2 5 0 2

Sample Output

Length:8

Elements:23 2 2 344 3 32 2 5

Count of deleted elements:3

Length:5

Elements:3 5 23 32 344

#include<iostream>

#include <cmath>

using namespace std;

const int MaxSize=100;  //数组中元素的最大个数

class Array

{

public:

    Array ();      //初始化length=0

    int Length();   //返回数组中有效元素个数

    void Insert(int i, double x);  //在下标i处插入x

    void Display();       //输出数组中有效元素

    void Sort();  //排序

    int  Delete(double x); //删除

private:

    double data[MaxSize];  //存储元素

    int length;              //数组中有效元素个数

};

Array::Array()

{

    length=0;

}

int Array::Length()

{

    return  length;

}

void Array::Insert(int i,double x)

{   int z;

if(i<0||i>length) throw "illegal position";

        else

  for(z=length-1;z>i;z--)

      data[z+1]=data[z];

  data[i]=x;

   length++;

}

int Array::Delete(double x)

{    int i=0,z,n=0;

   for(;i<length;i++)

       if(data[i]==x)

       {for(z=i;z<length-1;z++)

       data[z]=data[z+1];i--;length--;n++;}

       return n;

}

void Array::Display()

{  int i;

cout<<"Length:"<<length<<endl;

cout<<"Elements:";

for(i=0;i<length;i++) cout<<data[i]<<" ";

    cout<<endl;

}

void Array::Sort()

{   int i,z,t;

    for(i=length-1;i>0;i--)

        for(z=i-1;z>=0;z--)

   if(data[z]>data[i])

   {t=data[z];data[z]=data[i];data[i]=t;}

}

int main()

{   int i=0;

    double num;

    Array a; //顺序表的初始化

    while(1)

    {

    cin>>num;

    if(!num) break;

    try{

    a.Insert(i++,num);  //insert

    }

    catch(char \*ms)

    {

    cout<<ms<<endl;

    }

    }

    a.Display ();

    double x;

    cin>>x;

    try

    {

    int n=a.Delete(x);   //delete all x

    cout<<"Count of deleted elements:"<<n<<endl;

    }

    catch(char \*ms)

    {

    cout<<ms<<endl;

    }

    a.Sort();   //sort

    a.Display();

    return 0;

}

## 1187: 类与对象14：Array类3

Time Limit: 1 Sec  Memory Limit: 128 MB  
Submit: 122  Solved: 87  
[[Submit](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/submitpage.php?id=1187)][[Status](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/problemstatus.php?id=1187)][[Web Board](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/bbs.php?pid=1187)]

Description

数组类Array用于存储一组双精度浮点数，其定义如下，给出各个成员函数实现。

class Array

{

public:

Array (int size=100); //初始化:MaxSize置为size，为data动态分配内存，length置为0

Array (const Array &a); //深拷贝构造函数

~Array (); //析构函数

void Insert(int i, double x); //在下标i处插入x

void Display(); //输出数组中的元素

private:

int MaxSize; //数组中可存储的元素最大个数

double \*data; //存储元素

int length; //数组中有效元素个数

};

main函数中输入n个(n<100)双精度浮点数（以0结束），将这n个数存储到同一个Array类对象arr1中，并调用Display输出这些数，利用arr1构造Array类对象arr2，并输出arr2中所有元素的值。

Input

Output

Sample Input

23 42 13 42 4 25 34 0

Sample Output

Length:7

Elements:23 42 13 42 4 25 34

Length:7

Elements:23 42 13 42 4 25 34

#include <iostream>

using namespace std;

class Array

{

public:

    Array(int size=100);

    Array(const Array &a);

    ~Array();

    void Insert(int i,double x);

    void Display();

private:

    int MaxSize;

    double \*data;

    int length;

};

Array::Array(int size)

{

    MaxSize=size;

    data=new double[MaxSize];

    length=0;

}

Array::Array(const Array &a)

{   int i;

    length=a.length;

    data=new double[a.length];

    for(i=0;i<length;i++)

        data[i]=a.data[i];

}

Array::~Array()

{

    delete data;

}

void Array::Insert(int i,double x)

{

    int z;

    for(z=length-1;z>=i;z--)

        data[z+1]=data[z];

    data[i]=x;

    length++;

}

void Array::Display()

{   int i;

     cout<<"Length:"<<length<<endl;

    cout<<"Elements:";

    for(i=0;i<length;i++)

        cout<<data[i]<<" ";

        cout<<endl;

}

int main()

{

    int i=0,n,x;

    Array arr1;

    while(cin>>x)

    {   if(x==0)break;

        arr1.Insert(i++,x);

    }

    arr1.Display();

    Array arr2(arr1);

    arr2.Display();

    return 0;

}

## 1188: 类与对象15：Array类4

Time Limit: 1 Sec  Memory Limit: 128 MB  
Submit: 181  Solved: 73  
[[Submit](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/submitpage.php?id=1188)][[Status](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/problemstatus.php?id=1188)][[Web Board](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/bbs.php?pid=1188)]

Description

数组类Array用于存储一组双精度浮点数，其定义如下，给出各个成员函数实现。

class List

{

public:

Array (int size=100); //初始化数据成员：MaxSize置为size，为data动态分配内存空间，length置为0

Array (const Array&r);//深拷贝构造函数

~Array (); //析构函数

void Insert(int i, double x); //在下标i处插入x

void Display(); //输出数组中的元素

private:

int MaxSize; //数组中可存储的元素最大个数

double \*data; //存储元素

int length; //数组中有效元素个数

};

为Array类增加下列成员函数：

（1）查找位置Locate

功能：在数组中查找x，若存在则返回x在数组中的下标，若不存在则返回-1。假设x至多存在一个。

（2）逆置函数Invert

功能：将data中的各元素逆置存放。

main函数中输入n个(n<100)双精度浮点数（以0结束），将这n个数存储到同一个Array类对象arr1中，并调用Display输出这些数，再验证新增成员函数功能。

Input

Output

Sample Input

12 42 535 66 64 2 4 0 535

Sample Output

Length:7

Elements:12 42 535 66 64 2 4

Position of 535:2

Length:7

Elements:4 2 64 66 535 42 12

#include <iostream>

using namespace std;

class Array

{

public:

    Array(int size=100);

    Array(const Array &a);

    ~Array();

    void Insert(int i,double x);

    void Display();

    void Invert();

    int Locate(double x);

private:

    int MaxSize;

    double \*data;

    int length;

};

int Array::Locate(double x)

{

    int i;

    for(i=0;i<length;i++)

        if(data[i]==x) return i;

    if(i==length) return -1;

}

void Array::Invert()

{

  int i=0;

  double t;

  for(i=0;i<=length/2-1;i++)

    {t=data[length-i-1];

    data[length-i-1]=data[i];

    data[i]=t;}

}

Array::Array(int size)

{

    MaxSize=size;

    data=new double[MaxSize];

    length=0;

}

Array::Array(const Array &a)

{   int i;

    length=a.length;

    data=new double[a.length];

    for(i=0;i<length;i++)

        data[i]=a.data[i];

}

Array::~Array()

{

    delete data;

}

void Array::Insert(int i,double x)

{

    int z;

    for(z=length-1;z>=i;z--)

        data[z+1]=data[z];

    data[i]=x;

    length++;

}

void Array::Display()

{   int i;

     cout<<"Length:"<<length<<endl;

    cout<<"Elements:";

    for(i=0;i<length;i++)

        cout<<data[i]<<" ";

        cout<<endl;

}

int main()

{

    int i=0,n,x;

    Array arr1;

    while(cin>>x)

    {   if(x==0)break;

        arr1.Insert(i++,x);

    }

    arr1.Display();

    cin>>x;

    cout<<"Position of "<<x<<": "<<arr1.Locate(x)<<endl;

    arr1.Invert();

    arr1.Display();

    return 0;

}

## 1189: 类与对象16：Point类4

Time Limit: 1 Sec  Memory Limit: 128 MB  
Submit: 177  Solved: 129  
[[Submit](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/submitpage.php?id=1189)][[Status](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/problemstatus.php?id=1189)][[Web Board](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/bbs.php?pid=1189)]

Description

定义一个点类Point，数据成员为平面点坐标x，y，成员函数有：构造函数，函数display显示点信息，函数GetX，GetY分别获取私有成员x，y的值。编写一个普通函数，计算两点间距离，函数原型double Distance(Point &p1,Point &p2)。主函数中输入两个点信息，定义两个类对象，利用display显示各自的点信息，再利用Distance计算两点距离。

Input

Output

Sample Input

0 0 10 20

Sample Output

Point(0,0)

Point(10,20)

Distance:22.3607

#include<iostream>

#include <cmath>

using namespace std;

class Point

{

private:

double x,y;

public:

Point (double x=0,double y=0);

double Getx();

double Gety();

void Display();

};

Point::Point (double x1,double y1)

{ x=x1;

y=y1;

}

double Point::Getx()

{

return x;

}

double Point::Gety()

{

return y;

}

void Point::Display()

{

cout<<"Point("<<x<<","<<y<<")"<<endl;

}

double distance(Point &p1,Point &p2)

{

return sqrt((p1.Getx()-p2.Getx())\*(p1.Getx()-p2.Getx())+

(p1.Gety()-p2.Gety())\*(p1.Gety()-p2.Gety()));

}

int main()

{

double x1,y1,x2,y2;

cin>>x1>>y1>>x2>>y2;

Point p1(x1,y1);

Point p2(x2,y2);

p1.Display();

p2.Display();

cout<<"Distance:"<<distance(p1,p2)<<endl;

return 0;

}

## 1190: 类与对象17：Point类5

Time Limit: 1 Sec  Memory Limit: 128 MB  
Submit: 7  Solved: 7  
[[Submit](http://221.203.21.203:8000/rewriter/USTL/http/489629629022/OnlineJudge/submitpage.php?id=1190)][[Status](http://221.203.21.203:8000/rewriter/USTL/http/489629629022/OnlineJudge/problemstatus.php?id=1190)][[Web Board](http://221.203.21.203:8000/rewriter/USTL/http/489629629022/OnlineJudge/bbs.php?pid=1190)]

Description

定义一个点类Point，数据成员为平面点坐标x，y，成员函数有：构造函数，函数display显示点信息，成员函数Distance实现计算两个点间距离。函数原型double Distance(Point &p)。主函数中输入两个点信息，定义两个类对象，利用display显示各自的点信息，再利用Distance计算两点距离。

Input

Output

Sample Input

0 0 10 20

Sample Output

Point(0,0)

Point(10,20)

Distance:22.3607

#include<iostream>

#include <cmath>

using namespace std;

class Point

{

 private:

     double x,y;

 public:

     Point (double x=0,double y=0);

         double Getx();

         double Gety();

          void Display();

};

Point::Point (double x1,double y1)

{   x=x1;

    y=y1;

}

double Point::Getx()

{

   return x;

}

double Point::Gety()

{

    return y;

}

void Point::Display()

{

    cout<<"Point("<<x<<","<<y<<")"<<endl;

}

double distance(Point p1,Point p2)

{

    return sqrt((p1.Getx()-p2.Getx())\*(p1.Getx()-p2.Getx())+

        (p1.Gety()-p2.Gety())\*(p1.Gety()-p2.Gety()));

}

int main()

{

    double x1,y1,x2,y2;

    cin>>x1>>y1>>x2>>y2;

     Point p1(x1,y1);

     Point p2(x2,y2);

     p1.Display();

     p2.Display();

    cout<<"Distance:"<<distance(p1,p2)<<endl;

    return 0;

}

## 1191: 类与对象18：Point类6

Time Limit: 1 Sec  Memory Limit: 128 MB  
Submit: 7  Solved: 7  
[[Submit](http://221.203.21.203:8000/rewriter/USTL/http/489629629022/OnlineJudge/submitpage.php?id=1191)][[Status](http://221.203.21.203:8000/rewriter/USTL/http/489629629022/OnlineJudge/problemstatus.php?id=1191)][[Web Board](http://221.203.21.203:8000/rewriter/USTL/http/489629629022/OnlineJudge/bbs.php?pid=1191)]

Description

定义一个点类Point，数据成员为平面点坐标x，y，成员函数有：构造函数，函数display显示点信息，利用友元函数Distance实现计算两个点间距离。函数原型double Distance(const Point &p1,const Point &p2)。主函数中输入两个点信息，定义两个类对象，利用display显示各自的点信息，再利用Distance计算两点距离。

Input

Output

Sample Input

0 0 10 20

Sample Output

Point(0,0)

Point(10,20)

Distance:22.3607

#include<iostream>

#include <cmath>

using namespace std;

class Array

{

public:

Array(int size=100);

Array (const Array &a); //初始化length=0

void Insert(int i, double x); //在下标i处插入x

void Display(); //输出数组中有效元素

~Array();

private:

int MaxSize; //存储元素

int length;//数组中有效元素个数

double \*data;

};

Array::~Array()

{

delete data;

}

Array::Array(int size)

{

MaxSize=size;

length=0;

data=new double[size];

}

Array::Array(const Array &a)

{

int i=0;

length=a.length;

data=new double[a.MaxSize];

for(i=0;i<length;i++)

{

data[i]=a.data[i];

}

}

void Array::Insert(int i,double x)

{ int z;

for(z=length-1;z>i;z--)

data[z+1]=data[z];

data[i]=x;

length++;

}

void Array::Display()

{ int i;

cout<<"Length:"<<length<<endl;

cout<<"Elements:";

for(i=0;i<length;i++) cout<<data[i]<<" ";

cout<<endl;

}

int main()

{ int i=0;

double num;

Array arr1; //顺序表的初始化

while(1)

{

cin>>num;

if(!num) break;

arr1.Insert(i++,num);

}

arr1.Display ();

Array arr2(arr1);

arr2.Display();

return 0;

}

## 1192: 类与对象19：Complex类2

Time Limit: 1 Sec  Memory Limit: 128 MB  
Submit: 75  Solved: 67  
[[Submit](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/submitpage.php?id=1192)][[Status](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/problemstatus.php?id=1192)][[Web Board](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/bbs.php?pid=1192)]

Description

定义一个复数类Complex，该类对象存放一个复数的实部和虚部。设计带有默认形参值的构造函数，实部虚部的默认值均为0；函数ComAdd实现复数加法运算（以成员函数方式实现两种，普通函数方式实现一种）；设计一个能够输出复数的Display成员函数（当虚部为0时仅输出实部；输出样式如：18.5、10+3i、10-3.5i、-5.8+9.5i、-4.5-9.6i等）。

类的定义如下：

class Complex

{

private:

       double real,imag;

public:

       Complex(double r=0,double i=0);

       void Display();

       const double GetReal()

       {

              return real;

       }

       const double GetImag()

       {

              return imag;

       }

       Complex ComAdd(const Complex &c); //复数加法运算1

       Complex ComAdd(const Complex &c1,const Complex &c2);// 复数加法运算2

};

普通函数声明如下：

Complex ComAdd(Complex &c1,Complex &c2); // 复数加法运算3

main函数如下(不得修改main函数)：

int main()

{

       double r1,i1,r2,i2;

       cin>>r1>>i1;

       cin>>r2>>i2;

       Complex c1(r1,i1),c2(r2,i2),c3;

       c1.Display();

       c2.Display();

       c3=c1.ComAdd(c2); //成员函数实现复数加法运算方式1

       c3.Display();

       c3=c1.ComAdd(c1,c2); //成员函数实现复数加法运算方式2

       c3.Display();

    c3=ComAdd(c1,c2);   //普通函数实现复数加法运算方式3

       c3.Display();

return 0;

}

Input

Output

Sample Input

3 -5.5 6 3

Sample Output

3-5.5i

6+3i

9-2.5i

9-2.5i

9-2.5i

#include <iostream>

using namespace std;

class Complex

{

private:

       double real,imag;

public:

       Complex(double r=0,double i=0);

       void Display();

       const double GetReal()

       {return real;}

       const double GetImag()

       {return imag;}

       Complex ComAdd(const Complex &c);

       Complex ComAdd(const Complex &c1,const Complex &c2);

};

Complex ComAdd(Complex &c1,Complex &c2);

Complex::Complex(double r,double i)

{

  real=r;

  imag=i;

}

Complex Complex::ComAdd(const Complex &c)

{    double a,b;

    a=real+c.real;

    b=imag+c.imag;

    Complex c1(a,b);

    return c1;

}

Complex Complex::ComAdd(const Complex &c1,const Complex &c2)

{    Complex c;

    c.real=c1.real+c2.real;

    c.imag=c1.imag+c2.imag;

    return c;

}

void Complex::Display()

{

  if(imag>0) cout<<real<<"+"<<imag<<"i"<<endl;

  else if(imag<0) cout<<real<<imag<<"i"<<endl;

  else cout<<real<<endl;

}

Complex ComAdd(Complex &c1,Complex &c2)

 {

     double a,b;

    a=c1.GetReal()+c2.GetReal();

    b=c1.GetImag()+c2.GetImag();

    Complex c(a,b);

    return c;

    }

 int main()

{

       double r1,i1,r2,i2;

       cin>>r1>>i1;

       cin>>r2>>i2;

       Complex c1(r1,i1),c2(r2,i2),c3;

       c1.Display();

       c2.Display();

       c3=c1.ComAdd(c2);

       c3.Display();

       c3=c1.ComAdd(c1,c2); //成员函数实现复数加法运算方式2

       c3.Display();

       c3=ComAdd(c1,c2);   //普通函数实现复数加法运算方式3

       c3.Display();

return 0;

}

## 1193: 类与对象20：Date类

Time Limit: 1 Sec  Memory Limit: 128 MB  
Submit: 29  Solved: 17  
[[Submit](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/submitpage.php?id=1193)][[Status](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/problemstatus.php?id=1193)][[Web Board](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/bbs.php?pid=1193)]

Description

设计一个日期Date类，它能存储日期的年、月和日，Date类对象可在构造时初始化成员、也可在对象创建后改变数据成员；可以设定日期显示格式，当显示格式为“YY-MM-DD”时按照“年-月-日”格式显示，如2014-11-20，当显示格式为“MM-DD-YY”时按照“月-日-年”格式显示，如11-20-2014，日期显示格式以静态成员形式存储；该类可实现在对象当前日期基础上增加n天（n>0表示增加n天，n<0表示减小n天，注意：增加时可能引起月份和年份的变化）。

Main函数如下：（不得修改main函数）

int main()

{

int y1,m1,d1;

int y2,m2,d2;

int dx;

char SDF[10];

cin>>y1>>m1>>d1;

cin>>y2>>m2>>d2;

cin>>dx;

cin>>SDF; //输入日期显示格式

Date dt1;

dt1.Display();

dt1.SetDate(y1,m1,d1);

dt1.Display();

Date dt2(y2,m2,d2);

dt2.Display();

Date::SetDateFormat(SDF); //设置日期显示格式

dt1.Display();

dt2.Display();

dt2.AddDay(dx);

dt2.Display();

return 0;

}

Input

Output

Sample Input

2012 10 12

2011 2 1

36

MM-DD-YY

Sample Output

2014-1-1

2012-10-12

2011-2-1

10-12-2012

2-1-2011

3-9-2011

#include <iostream>

#include<cstring>

using namespace std;

int day\_num[13]={0,31,28,31,30,31,30,31,31,30,31,30,31};

class Date

{

private:

int year;

int month;

int day;

static char Format[9];

public:

Date(int y=2014,int m=1,int d=1):year(y),month(m),day(d){};

~Date(){};

void SetDate(int y,int m,int d){

year=y;

month=m;

day=d;

}

void AddDay(int add);

void Display();

static void SetDateFormat(char SDF[10]){

strcpy(Format,SDF);

}

};

char Date::Format[9]="YY-MM-DD";

void Date::AddDay(int add)

{

while(add>0){

day\_num[2]=((year%100&&(year%400==0))||year%4==0)?29:28;

for(;month<13&&add+day>day\_num[month];day=1,month++)

add-=(day\_num[month]-day+1);

month%=12;

if(add+day<=day\_num[month]){

day+=add;break;

}

year++;

}

while(add<0){

day\_num[2]=((year%100&&(year%400==0))||year%4==0)?29:28;

while(add+day<=0){

month=month==1?12:month-1;

add+=day;

day=day\_num[month];

if(month==12) break;

}

if(add+day>0){

day+=add;

break;

}

year--;

}

}

void Date::Display()

{

if(!strcmp(Format,"YY-MM-DD"))

cout<<year<<"-"<<month<<"-"<<day<<endl;

else cout<<month<<"-"<<day<<"-"<<year<<endl;

}

int main()

{

int y1,m1,d1;

int y2,m2,d2;

int dx;

char SDF[10];

cin>>y1>>m1>>d1;

cin>>y2>>m2>>d2;

cin>>dx;

cin>>SDF; //输入日期显示格式

Date dt1;

dt1.Display();

dt1.SetDate(y1,m1,d1);

dt1.Display();

Date dt2(y2,m2,d2);

dt2.Display();

Date::SetDateFormat(SDF); //设置日期显示格式

dt1.Display();

dt2.Display();

dt2.AddDay(dx);

dt2.Display();

return 0;

}

1194: 类与对象21：String类1

Time Limit: 1 Sec  Memory Limit: 128 MB  
Submit: 132  Solved: 93  
[[Submit](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/submitpage.php?id=1194)][[Status](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/problemstatus.php?id=1194)][[Web Board](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/bbs.php?pid=1194)]

Description

定义一个String类，数据成员包括私有数据成员：char \*str;成员函数包括构造函数String（用于实现对数据成员初始化），成员函数Display（用于输出串信息），求串长函数，拷贝构造函数（利用已知对象创建新对象）。编写主函数，定义对象S1，输出对象信息和串长信息；再利用拷贝构造函数创建对象S2，输出对象S2的信息。

输入数据为串信息，利用构造函数传递给数据成员str。

Input

Output

Sample Input

China

Sample Output

China

5

China

#include<iostream>

using namespace std;

#include <cstring>

class String

{

private:

char \*str;

public:

String(char \*str1);

void Display();

void Len();

String(const String &s1);

};

String::String(const String &s1)

{

int len;

len=strlen(s1.str);

str=new char [len+1];

strcpy(str,s1.str);

}

void String::Display()

{

cout<<str<<endl;

}

String::String(char \*str1)

{

int len;

len=strlen(str1);

str=new char [len+1];

strcpy(str,str1);

}

void String::Len()

{ int i;

for(i=0;str[i];i++);

cout<<i<<endl;

}

int main()

{

char s[20];

cin>>s;

String s1(s);

s1.Display();

s1.Len();

String s2(s1);

s2.Display();

return 0;

}

## 1195: 类与对象22：String类2

Time Limit: 1 Sec  Memory Limit: 128 MB  
Submit: 97  Solved: 66  
[[Submit](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/submitpage.php?id=1195)][[Status](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/problemstatus.php?id=1195)][[Web Board](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/bbs.php?pid=1195)]

Description

定义一个String类，数据成员包括私有数据成员：char \*str;成员函数包括构造函数String（用于实现对数据成员初始化），成员函数Display（用于输出串信息），串比较函数(返回第一个不同字符的ACSII码差值)，串连接函数和串拷贝函数。

主函数如下，依次测试上述成员函数。

int main()

{

char str1[81],str2[81];

cin>>str1>>str2;

CString s1(str1);

s1.Display();

CString s2(str2);

s2.Display();

cout<<s2.MyStrcmp(s1)<<endl;

s2.MyStrcpy(s1);

s2.Display();

s2.MyStrcat(s1);

s2.Display();

}

Input

Output

Sample Input

China Chinese

Sample Output

China

Chinese

4

China

ChinaChina

#include<iostream>

using namespace std;

#include <cstring>

class CString

{

private:

char \*str;

public:

CString(char \*str1);

void Display();

int MyStrcmp(CString &s1);

void MyStrcpy(const CString &s1);

void MyStrcat(CString &s1);

};

int CString::MyStrcmp(CString &s1)

{ int i;

for(i=0;str[i];i++)

if(str[i]!=s1.str[i]) break;

return str[i]-s1.str[i];

}

void CString::MyStrcat(CString &s1)

{ int i,z;

for(i=0;str[i];i++);

for(z=0;s1.str[z];z++)

str[i++]=s1.str[z];

str[i]=0;

}

void CString::MyStrcpy(const CString &s1)

{

int len;

len=strlen(s1.str);

str=new char [len+1];

strcpy(str,s1.str);

}

void CString::Display()

{

cout<<str<<endl;

}

CString::CString(char \*str1)

{

int len;

len=strlen(str1);

str=new char [len+1];

strcpy(str,str1);

}

int main()

{

char str1[81],str2[81];

cin>>str1>>str2;

CString s1(str1);

s1.Display();

CString s2(str2);

s2.Display();

cout<<s2.MyStrcmp(s1)<<endl;

s2.MyStrcpy(s1);

s2.Display();

s2.MyStrcat(s1);

s2.Display();

}

## 1196: 类与对象23：Complex类3

Time Limit: 1 Sec  Memory Limit: 128 MB  
Submit: 95  Solved: 63  
[[Submit](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/submitpage.php?id=1196)][[Status](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/problemstatus.php?id=1196)][[Web Board](http://221.203.21.203:8001/rwt/USTL/http/GV6T6N3UFZ5UGLSSGM3B/OnlineJudge/bbs.php?pid=1196)]

Description

定义一个复数类Complex，该类对象存放一个复数的实部和虚部。设计带有默认形参值的构造函数，实部虚部的默认值均为0；设计一个能够输出复数的Display成员函数（当虚部为0时仅输出实部、当实部为0时仅输出虚部，若实部与虚部同时为0时，输出0；输出样式如：-20、18.5、10+3i、9i、-9i、10-3.5i、-5.8+9.5i、-4.5-9.6i等）。

main函数如下(不得修改main函数)：

int main()

{

double r1,i1,r2,i2;

cin>>r1>>i1;

cin>>r2>>i2;

Complex c1(r1,i1);

Complex c2;

c1.Display();

c2.Display();

c2.Set(r2,i2);

c2.Display();

return 0;

}

程序运行时输入：

0 -5 10 3

则程序运行输出为：

-5i

0

10+3i

Input

Output

Sample Input

-5 0

10 -9

Sample Output

-5

0

10-9i

#include <iostream>

using namespace std;

class Complex

{

private:

       double real,imag;

public:

       Complex(double r=0,double i=0);

       void Display();

       void Set(double r1,double i1) ;

};

void Complex::Set(double r1,double i1)

{

    real=r1;

    imag=i1;

}

Complex::Complex(double r,double i)

{

  real=r;

  imag=i;

}

void Complex::Display()

{

  if(imag>0&&real!=0) cout<<real<<"+"<<imag<<"i"<<endl;

  else if(imag>0&&real==0) cout<<"+"<<imag<<"i"<<endl;

  else if(imag<0&&real!=0) cout<<real<<imag<<"i"<<endl;

  else if(imag<0&&real==0) cout<<imag<<"i"<<endl;

  else cout<<real<<endl;

}

int main()

{

    double r1,i1,r2,i2;

    cin>>r1>>i1;

    cin>>r2>>i2;

    Complex c1(r1,i1);

    Complex c2;

    c1.Display();

    c2.Display();

    c2.Set(r2,i2);

    c2.Display();

    return 0;

}