数据结构实验：复数的四则运算

#include <iostream.h>

#include <math.h>

typedef struct complex

{

float real;

float imag;

}complex;

void initcomplex(complex&z)

{

cout<<"输入复数的实部:"<<endl;

cin>>z.real ;

cout<<"输入复数的虚部:"<<endl;

cin>>z.imag;

}

void outputcomplex(complex z)

{

if(z.real!=0)

{

cout<<z.real;

if(z.imag>0)

cout<<"+"<<z.imag<<"i"<<endl;

else if(z.imag==0)

cout<<endl;

else if(z.imag<0)

cout<<z.imag<<"i"<<endl;

}

if(z.real==0&&z.imag>0)

cout<<z.imag<<"i"<<endl;

if(z.real==0&&z.imag==0)

cout<<z.imag<<endl;

if(z.real==0&&z.imag<0)

cout<<z.imag<<"i"<<endl;

}

void addcomplex(complex &x,complex y, complex z)

{

x.real=y.real+z.real;

x.imag=y.imag+z.imag;

}

void minuscomplex(complex &x,complex y,complex z)

{

x.real=y.real-z.real;

x.imag=y.imag-z.imag;

}

void mulcomplex(complex &x,complex y,complex z)

{

x.real=y.real\*z.real-y.imag\*z.imag;

x.imag=y.real\*z.imag+y.imag\*z.real;

}

void ericomplex(complex &x,complex y,complex z)

{

x.real=(y.real\*z.real+y.imag\*z.imag)/(z.real\*z.real+z.imag\*z.imag);

x.imag=(y.imag\*z.real-y.real\*z.imag)/(z.real\*z.real+z.imag\*z.imag);

}

void main()

{

complex z1,z2,z3;

initcomplex(z1);

cout<<"第一个复数:";

outputcomplex(z1);

initcomplex(z2);

cout<<"第二个复数:";

outputcomplex(z2);

addcomplex(z3,z1,z2);

cout<<"两个复数的和:";

outputcomplex(z3);

minuscomplex(z3,z1,z2);

cout<<"两个复数的差:";

outputcomplex(z3);

ericomplex(z3,z1,z2);

cout<<"两个复数的商:";

outputcomplex(z3);

if (z1.real!=0||z2.real!=0)

{

mulcomplex(z3,z1,z2);

cout<<"两个复数的积:";

outputcomplex(z3);

}

if (z1.real==0&&z2.real==0)

cout<<"两个复数的积:"<<"-"<<z1.imag\*z2.imag<<endl;

}