程序代码：

#include <iostream>

using namespace std;

class matrixmul

{

private:

double real;

double imag;

public:

void output();//打印输出

void initSet(double re,double im);//初始化

matrixmul matrixMul(matrixmul Z1,matrixmul Z2);//函数返回值是multirxmul，所以此函数类型为matrixmal

};

matrixmul matrixmul::matrixMul(matrixmul Z1,matrixmul Z2)

{

double temp1,temp2,temp3;

matrixmul result;

temp1=Z1.real\*Z2.imag;

temp2=Z1.imag\*Z2.real;

temp3=(Z1.imag+Z1.real)\*(Z2.real-Z2.imag);

result.real=temp1+temp3-temp2;

result.imag=temp1+temp2;

return result;

}

void matrixmul::initSet(double re,double im)

{

real=re;

imag=im;

}

void matrixmul::output()

{

if(imag>0)

{

cout<<real<<"+"<<imag<<"i"<<endl;

}

else if(imag==0)

{

cout<<real<<endl;

}

else if(imag<0)

{

cout<<real<<imag<<"i"<<endl;

}

}

int main()

{

matrixmul Z1,Z2,Z3,result;

Z1.initSet(1,-2);

Z2.initSet(3,-4);

cout<<"第一个复数Z1为："<<endl;

Z1.output();

cout<<"第二个复数Z2为："<<endl;

Z2.output();

result=Z3.matrixMul(Z1,Z2);

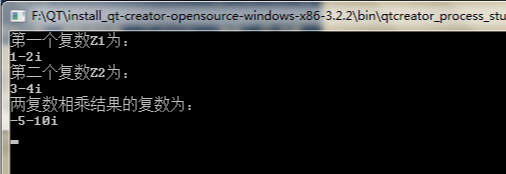
cout<<"两复数相乘结果的复数为："<<endl;

result.output();

return 0;

}

运行结果：



感想：

设计算法，仅使用三次实数乘法即可完成复数Z1=a+b\*i,Z2=c+d\*i相乘，根据复数乘法的计算公式：Z1\*Z2=（a\*c-b\*d）+(a\*d+b\*c)i，可以使用加法或者减法来减少乘法运算的时间。计算temp1=a\*d,temp2=b\*c,temp3=(a+b)(c-d)，虚部为temp1+temp2，实部为temp3+temp1-temp2

Copyright ©2021-2099 Wenhuiyu. All rights reserved