#include<iostream>

//#include<cstring>

#include<cmath>

using namespace std;

class complex {

private:

double real;

double image;

public:

complex(double real1=0 ,double image1=0):real(real1),image(image1){}

friend complex operator\*(complex& x, complex& y);

void show();

};

void complex::show()

{

cout << "The multiplication of the complexes is : " << endl;

cout << real;

if (image > 0)

cout << "+";

if(image!=0)

cout << image << "i" << endl;

}

complex operator\*(complex& x, complex& y)

{

complex temp;

temp.real = x.real \* y.real;

temp.image = x.image \* y.image;

return temp;

}

int main()

{

double x1, x2, y1, y2;

cout << "Please input the first complex: " << endl;

cin >> x1 >> y1;

cout << "Pease input the second complex: " << endl;

cin >> x2 >> y2;

complex a(x1 ,y1);

complex b(x2, y2);

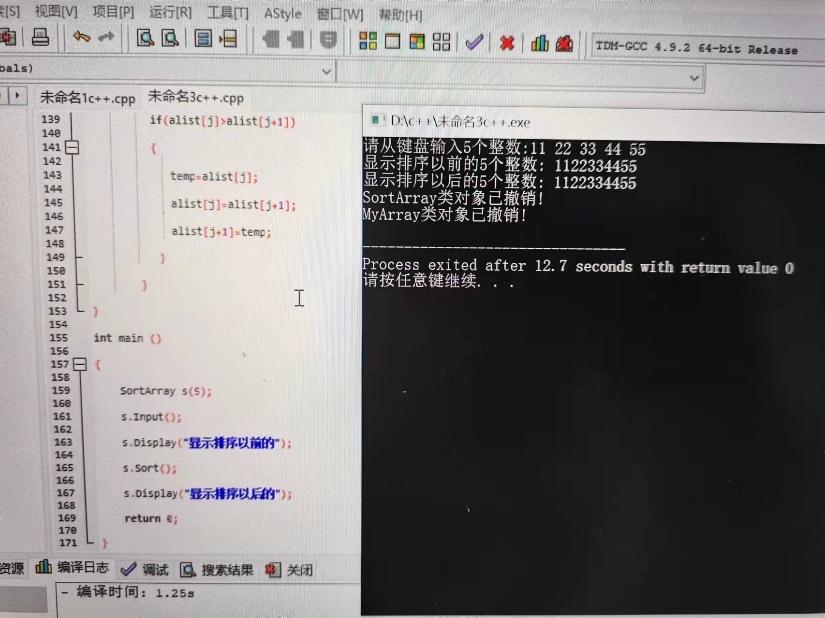
complex c;

c = a \* b;

c.show();

return 0;

}

图片：  


#include<iostream>

//#include<cstring>

#include<cmath>

using namespace std;

class complex {

private:

double real;

double image;

public:

complex(double real1=0 ,double image1=0):real(real1),image(image1){}

friend complex operator\*(complex& x, complex& y);

void show();

};

void complex::show()

{

cout << "The multiplication of the complexes is : " << endl;

cout << real;

if (image > 0)

cout << "+";

if(image!=0)

cout << image << "i" << endl;

}

complex operator\*(complex& x, complex& y)

{

complex temp;

temp.real = x.real \* y.real-x.image\*y.image;

temp.image = x.real \* y.image+x.image\*y.real;

return temp;

}

int main()

{

double x1, x2, y1, y2;

cout << "Please input the first complex: " << endl;

cin >> x1 >> y1;

cout << "Pease input the second complex: " << endl;

cin >> x2 >> y2;

complex a(x1 ,y1);

complex b(x2, y2);

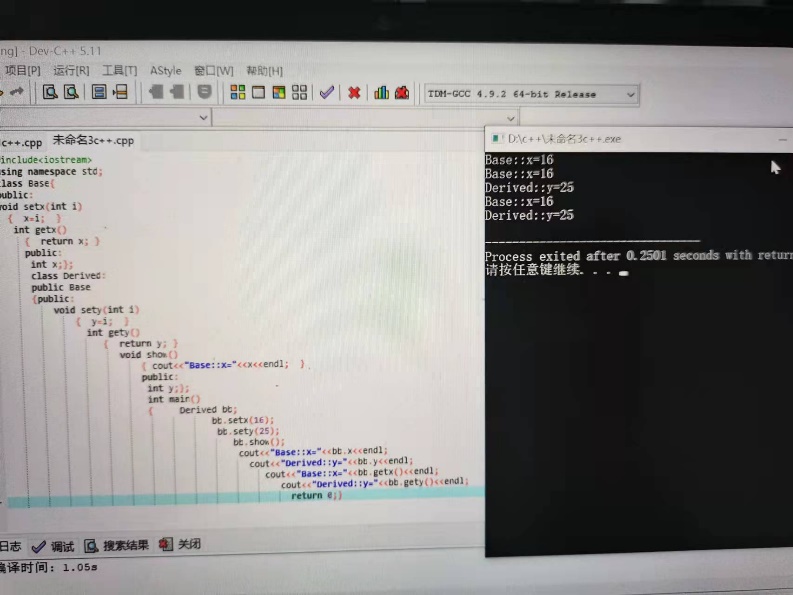
complex c;

c = a \* b;

c.show();

return 0;

}

图片：

心得：应用结构体知识，其中，因为输入结构体比较繁琐，所以建议使用函数来代替直接输入。其次，因为结构体一般用于制作表格，所以结构体数组是一个非常重要的运用。在实验过程中，我遇到了许多小错误，比如说简单的忘记定义结构体、不能整体赋值等，在实验过程中debug也变多了，锻炼了编写程序的耐心。