//Base.h

#include <iostream>

using namespace std;

class Base{

public:

virtual void show(){ cout<<"Base"<<endl;}

}

//Drived.h

#include "Base.h"

class Derived:public Base{

public:

void show(){ cout<<"Derived"<<endl; }

}

//main.c

#include <iostream>

#include <cmath>

#include <string>

#include "Base.h" //多文档

#include "Drived.h"

using namespace std;

//考点 运算符重载

class Complex{

private:

int real, image;

public:

Complex(int r, int i):real(r),image(i){}

Complex operator+(Complex& c)

{

return Complex(real+c.real,image+c.image);

}

float GetM();

};

float Complex::GetM(){

return sqrt(real\*real+image\*image);

}

//编程题考点 类的继承，请补充完整类的定义（构造函数等）

Point\_1{

private:

int x;

};

Point\_2:public Point\_1{

private:

int y;

};

Point\_3:public Point\_2{

private:

int z;

int r;

};

int main(){

//读程序题 考点 指针，地址，变量

int i;

int \*pi;

i=30;

pi=&i;

cout<<i<<\*pi<<endl;

int data[5]={3,4,6,1,9};

int \*pdata=data+1;

int a=\*(pdata+1); //6

int b=pdata[3]; //9

Complex c1(3,4),c2(5,6);

Complex c3=c1+c2;

cout<<c3.GetM();

cout<<c2.GetM();

//考点：库函数 strlen

char str[20]="hello";

string str="hello";

str.length(); //size\_t length(const char \*p)

//多文档，静态连遍，动态连遍

Base b1, \*pb;

Derived d1;

pb=&d1;

b1.show(); //base

d1.show(); //Derived

pb.show(); //b 静态连遍

}