#include <iostream>

using namespace std;

class Complex {

private:

int real, im;

public:

Complex() {

real=0;

im=0;

}

Complex(int r, int i) {

real = r;

im = i;

}

Complex operator+(const Complex& c) { // Use const reference to avoid unnecessary copying

Complex obj;

obj.real = real + c.real;

obj.im = im + c.im;

return obj;

}

friend istream& operator>>(istream& in, Complex& obj);

friend ostream& operator<<(ostream& out, const Complex& obj); // Pass by const reference

};

istream& operator>>(istream& in, Complex& obj) {

cout << "Enter the real part: ";

in >> obj.real;

cout << "Enter the imaginary part: ";

in >> obj.im;

return in;

}

ostream& operator<<(ostream& out, const Complex& obj) { // Use const reference

out << obj.real << " + " << obj.im << "i"; // Fixed typo: changed img to im

return out;

}

int main() {

Complex c1, c2, c3;

cin >> c1;

cin >> c2;

c3 = c1 + c2;

cout << "Sum is: " << c3 << endl;

return 0;

}