#include <iostream>

#include <fstream> //for file handling

#include <sstream> //for use of another string

#include <string>

#include <cmath> //for use of operations

using namespace std;

class ComplexNumber {

double real;

double imaginary;

public:

ComplexNumber() : real(0), imaginary(0) {} // Constructors

ComplexNumber(double re, double im) : real(re), imaginary(im) {}

ComplexNumber(const string& complexStr) {

stringstream ss(complexStr);

char ch;

ss >> ch >> real >> ch >> imaginary >> ch;

}

ComplexNumber(const ComplexNumber& other) : real(other.real), imaginary(other.imaginary) {}

ComplexNumber operator+(const ComplexNumber& other) const {

return ComplexNumber(real + other.real, imaginary + other.imaginary);

}

ComplexNumber operator-(const ComplexNumber& other) const {

return ComplexNumber(real - other.real, imaginary - other.imaginary);

}

ComplexNumber operator\*(const ComplexNumber& other) const {

return ComplexNumber(real \* other.real - imaginary \* other.imaginary,

real \* other.imaginary + imaginary \* other.real);

}

ComplexNumber operator/(const ComplexNumber& other) const {

double denominator = other.real \* other.real + other.imaginary \* other.imaginary;

if (denominator != 0) {

return ComplexNumber((real \* other.real + imaginary \* other.imaginary) / denominator,

(imaginary \* other.real - real \* other.imaginary) / denominator);

}

else {

cerr << "Division by zero is not defined in complex numbers." << endl;

return ComplexNumber(NAN, NAN);

}

}

double getReal() const { return real; }

double getImaginary() const { return imaginary; }

void print() const {

cout << "(" << real << "+" << imaginary << "i)";

}

};

int main() {

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "| |" << endl;

cout << "| |" << endl;

cout << "| Welcome to complex number Calculator |" << endl;

cout << "| |" << endl;

cout << "| |" << endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

ifstream inputFile("input.txt");

ofstream outputFile("output.txt");

if (!inputFile) {

cerr << "Error: Unable to open input file." << endl;

return 1;

}

if (!outputFile) {

cerr << "Error: Unable to create output file." << endl;

return 1;

}

string line;

int lineNumber = 1;

while (getline(inputFile, line)) {

stringstream ss(line);

string complexStr1, complexStr2;

char op;

ss >> complexStr1 >> op >> complexStr2;

ComplexNumber c1(complexStr1);

ComplexNumber c2(complexStr2);

ComplexNumber result;

switch (op) {

case '+':

result = c1 + c2;

break;

case '-':

result = c1 - c2;

break;

case '\*':

result = c1 \* c2;

break;

case '/':

result = c1 / c2;

break;

default:

cerr << "Error: Invalid operator '" << op << "'." << endl;

result = ComplexNumber(NAN, NAN);

}

outputFile << lineNumber << ". ";

outputFile << line << " = ";

outputFile << result.getReal() << "+" << result.getImaginary() << "i" << endl;

cout << lineNumber << ". ";

cout << line << " = ";

cout << result.getReal() << "+" << result.getImaginary() << "i" << endl;

lineNumber++;

}

inputFile.close();

outputFile.close();

}