**算法训练 P1103**

时间限制：1.0s   内存限制：256.0MB

﻿  
　　编程实现两个复数的运算。设有两个复数 和 ，则他们的运算公式为：  
  
　　要求：（1）定义一个结构体类型来描述复数。  
　　（2）复数之间的加法、减法、乘法和除法分别用不用的函数来实现。  
　　（3）必须使用结构体指针的方法把函数的计算结果返回。  
　　说明：用户输入：运算符号(+,-,\*,/) a b c d.  
　　输出：a+bi，输出时不管a,b是小于0或等于0都按该格式输出，输出时a,b都保留两位。  
  
**输入：**  
　　- 2.5 3.6 1.5 4.9  
**输出：**  
　　1.00+-1.30i

本题的C++参考代码如下：

#include<iostream>

#include<iomanip>

using namespace std;

struct fushu

{

double a;

double b;

};

int main()

{

char t;

cin>>t;

fushu a,b;

cin>>a.a>>a.b;

cin>>b.a>>b.b;

switch(t)

{

case'+':

cout<<fixed<<setprecision(2)<<a.a+b.a<<"+"<<a.b+b.b<<"i"<<endl;

break;

case'-':

cout<<fixed<<setprecision(2)<<a.a-b.a<<"+"<<a.b-b.b<<"i"<<endl;

break;

case'\*':

cout<<fixed<<setprecision(2)<<a.a\*b.a-a.b\*b.b<<"+"<<a.b\*b.a+b.b\*a.a<<"i"<<endl;

break;

case'/':

{

double t;

t=b.a\*b.a+b.b\*b.b;

double x,y;

double p,q;

x=b.a;y=-b.b;

p=a.a\*x-a.b\*y;

q=a.b\*x+y\*a.a;

cout<<fixed<<setprecision(2)<<p/t<<"+"<<q/t<<"i"<<endl;

break;

}

}

}

本题的C参考代码如下：

#include<stdio.h>

struct number

{

float a;

float b;

}s[3];

void jia()

{

s[2].a=s[0].a+s[1].a;

s[2].b=s[0].b+s[1].b;

}

void jian()

{

s[2].a=s[0].a-s[1].a;

s[2].b=s[0].b-s[1].b;

}

void cheng()

{

s[2].a=s[0].a\*s[1].a-s[0].b\*s[1].b;

s[2].b=s[0].a\*s[1].b+s[0].b\*s[1].a;

}

void chu()

{

s[2].a=(s[0].a\*s[1].a+s[0].b\*s[1].b)/(s[1].a\*s[1].a+s[1].b\*s[1].b);

s[2].b=(-s[0].a\*s[1].b+s[0].b\*s[1].a)/(s[1].a\*s[1].a+s[1].b\*s[1].b);

}

int main()

{

char m;

scanf("%c %f %f %f %f",&m,&s[0].a,&s[0].b,&s[1].a,&s[1].b);

if(m=='+') jia(s[0].a,s[0].b,s[1].a,s[1].b);

if(m=='-') jian(s[0].a,s[0].b,s[1].a,s[1].b);

if(m=='\*') cheng(s[0].a,s[0].b,s[1].a,s[1].b);

if(m=='/') chu(s[0].a,s[0].b,s[1].a,s[1].b);

printf("%.2f+%.2fi",s[2].a,s[2].b);

return 0;

}

本题的Java参考代码如下：

import java.util.Scanner;

import java.text.DecimalFormat;

public class Main {

public static void main(String[] args){

Scanner in=new Scanner(System.in);

DecimalFormat df=new DecimalFormat("##0.00");

String gets=in.nextLine();

char op=gets.charAt(0);

double num[]=new double[4];

int i=2;

for(int k=0;k<4;k++){

num[k]=0;

double jinwei=1;

for(;i<gets.length()&&gets.charAt(i)!=' ';i++){

if(gets.charAt(i)!='.'){

if(jinwei>=1){

num[k]=num[k]\*jinwei+(gets.charAt(i)-'0');

jinwei\*=10;

}

else{

num[k]=num[k]+(gets.charAt(i)-'0')\*jinwei;

jinwei\*=0.1;

}

//System.out.println("num["+k+"]="+num[k]);

}

else{

jinwei=0.1;

}

}

i++;

}

if(op=='+'){

System.out.println(df.format(num[0]+num[2])+"+"+df.format(num[1]+num[3])+"i");

}

if(op=='-'){

System.out.println(df.format(num[0]-num[2])+"+"+df.format(num[1]-num[3])+"i");

}

if(op=='\*'){

System.out.println(df.format(num[0]\*num[2]-num[1]\*num[3])+"+"+df.format(num[0]\*num[3]+num[1]\*num[2])+"i");

}

if(op=='/'){

System.out.println(df.format((num[0]\*num[2]+num[1]\*num[3])/(num[3]\*num[3]+num[2]\*num[2]))+"+"+df.format((num[1]\*num[2]-num[0]\*num[3])/(num[3]\*num[3]+num[2]\*num[2]))+"i");

}

}

}