**复数归一化**

编写函数Normalize，将复数归一化，即若复数为a+bi，归一化结果为a/sqrt(a\*a+b\*b) + i\*b/sqrt(a\*a+b\*b) 。使用结构体指针类型作为函数参数可能是必要的。其中实部和虚部由键盘输入，输出为归一化结果，如果归一化结果的实部或虚部为小数的要求保留一位小数。

样例输入:（格式说明：3 4 分别为以空格隔开的实数的实部和虚部）   
3 4   
样例输出:  
0.6+0.8i

样例输入:   
2 5  
样例输出:  
0.4+0.9i

C++代码：

#include <stdio.h>

#include <math.h>

typedef struct complex{

float real, imag;

}COMPLEX, \*COMPLEXP;

int main(void){

float e;

COMPLEX com;

scanf("%f %f",&(com.real),&(com.imag));

e = sqrt(com.imag \* com.imag + com.real \* com.real);

com.imag = com.imag/e;

com.real = com.real/e;

printf("%0.1f+%0.1fi",com.real,com.imag);

return 0;

}

C代码：

#include <stdio.h>

#include <math.h>

//a/sqrt(a\*a+b\*b) + i\*b/sqrt(a\*a+b\*b)

int main()

{

int a,b;

scanf("%d%d",&a,&b);

printf("%.1f+%.1fi",a/sqrt(a\*a+b\*b),b/sqrt(a\*a+b\*b));

return 0;

}

JAVA代码：

import java.io.\*;

public class Main {

public static void main(String[] args) throws Exception {

// TODO Auto-generated method stub

BufferedReader buf=new BufferedReader(new InputStreamReader(System.in));

System.out.println(Normalize(buf.readLine()));

}

public static String Normalize(String str)

{

int a=0,b=0;

double c,d,e;

for(int i=0;i<str.length();i++)

{

if(str.charAt(i)==' ')

{

a=Integer.parseInt(str.substring(0,i));

b=Integer.parseInt(str.substring(i+1,str.length()));

break;

}

}

c=Math.sqrt(a\*a+b\*b);

d=a/c;

e=b/c;

d=(int)(d\*1000);

if(d%10>=5)

d=d+10;

if(d%100>=50)

d=d+100;

d=d/1000;

d=(int)(d\*10);

e=(int)(e\*1000);

if(e%10>=5)

e=e+10;

if(e%100>=50)

e=e+100;

e=e/1000;

e=(int)(e\*10);

str=(d/10+"+"+e/10+"i");

//a/sqrt(a\*a+b\*b) + i\*b/sqrt(a\*a+b\*b)

return str;

}

}