Quadratic Roots 1

Write a C program to print the Roots of a Quadratic Equation of the form ax^2+bx+c

Roots of the equation is given by the equation

Quadratic Formula

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

The nature of the roots are determined by the nature of the determinant

If determinant > 0,

$$root1 = \frac{-b + \sqrt{(b^2 - 4ac)}}{2a}$$

$$root2 = \frac{-b - \sqrt{(b^2 - 4ac)}}{2a}$$

If determinant = 0,

$$root1 = root2 = \frac{-b}{2a}$$

If determinant < 0,

$$root1 = \frac{-b}{2a} + i\frac{\sqrt{-(b^2 - 4ac)}}{2a}$$

$$root2 = \frac{-b}{2a} - i\frac{\sqrt{-(b^2 - 4ac)}}{2a}$$

Input Format

The user will input the coefficients of the quadratic equation a , b, c

First 3 numbers seperated by space will be a b and c respectively

Constraints

Only built in function that can be used is sqrt()

Output Format

Output should be the nature of the roots followed by the roots seperated by coma

Sample Input 0

```
1 2 5
```

Sample Output 0

```
Complex: -1.00 + i2.00 , -1.00 - i2.00
```

Sample Input 1

```
1 4 1
```

Sample Output 1

```
Distinct Real : -0.27 , -3.73
```

Sample Input 2

```
1 2 1
```

Sample Output 2

```
Equal Real: -1.00 , -1.00
```

Current Buffer (saved locally, editable)

C

```c
#include <stdio.h>
#include <math.h>

int main()
{
    int a,b,c;
    float d,real,img,root1,root2;
    scanf("%d%d%d",&a,&b,&c);

    d=(b*b)-(4*a*c);


    if(d<0)
    {
       real=(-b)/(float)(2*a);
       img=sqrt(-d)/(float)(2*a);

       printf("Complex: %0.2f + i%0.2f , %0.2f - i%0.2f ",real,img,real,img);
    }
    else if (d>0)
    {
        root1=(-b+sqrt(d))/(float)(2*a);
        root2=(-b-sqrt(d))/(float)(2*a);

```

```
        printf("Distinct Real : %0.2f , %.2f",root1,root2);
    }
    else
    {
        root1=root2=(-b)/(float)(2*a);
        printf("Equal Real: %0.2f , %.2f",root1,root2);
    }

   return 0;
}
```

Line: 1 Col: 1

Upload Code as File | Test against custom input | Run Code | Submit Code

Testcase 0 ✔ | Testcase 1 ✔ | Testcase 2 ✔

Congratulations, you passed the sample test case.

Click the **Submit Code** button to run your code against all the test cases.

Input (stdin)

```
1 2 5
```

Your Output (stdout)

```
Complex: -1.00 + i2.00 , -1.00 - i2.00
```

Expected Output

```
Complex: -1.00 + i2.00 , -1.00 - i2.00
```