assgn1_D1171708 Brian

```
printf("Enter coefficients a, b, and c for quadratic equation a x**2 + b x + c = 0: ");
scanf("%f %f %f", &a, &b, &c);
d = b*b - (4*a*c); //discriminant

if(d>0){
    e = r2c((-b + sqrt(b*b-4*a*c))/(2*a));
    f = r2c((-b - sqrt(b*b-4*a*c))/(2*a));
}

if(d==0){//r 2 c
    e = r2c((-b)/(2*a));
    f = r2c((-b)/(2*a));
}

if(d<0){
    e.re = (-b)/(2*a);
    e.im = sqrt(4*a*c-b*b)/(2*a);
    f.re = (-b)/(2*a);
    f.im = -sqrt(4*a*c-b*b)/(2*a);
}
```

In this assignment. Firstly, I use b*b-4*a*c to verify the square root is bigger than 0 or less than 0, or equal to 0, next, by using discriminant I divide the program into multiple real roots, two real roots, and two complex root parts.

```
typedef struct {
  float re; // real part
  float im;  // imaginary part
} complex;
```

Using the typedef struct can store the real part and the imaginary part of complex number.

```
complex r2c(float r){ //D == 0 real = ... im = 0
    complex c;
    c.re = r;
    c.im = 0;
    r`eturn c;
}
```

Also, converting the number which only contains real part to the complex number means the x.re = x , x.im = 0.

```
complex add(complex x, complex y){
    complex sum;
    sum.re = x.re + y.re;
    sum.im = x.im + y.im;
    return sum;
}

complex minus(complex x, complex y){
    complex sub;
    sub.re = x.re - y.re;
    sub.im = x.im - y.im;
    return sub;
}

complex time(complex x, complex y){
    complex prod;
    prod.re = x.re * y.re - x.im * y.im;
    prod.im = x.re * y.im + x.im * y.re;
    return prod;
}

complex divide(complex x, complex y){
    complex div;
    //((a×c+b×d)+(-a×d+b×c)i)÷(c2+d2)
    div.re = (x.re * y.re + x.im * y.im)/(y.re * y.re + y.im * y.im);
    div.im = (-x.re * y.im + x.im * y.re)/(y.re * y.re + y.im * y.im);

    return div;
}

float absComplex(complex x){
    float abs;
    abs = sqrt(x.re * x.re + x.im * x.im);
    return abs;
}
```

Complete the function given by the question.

```
void printComplex(complex r1, complex r2){
    if(r1.re != 0 && r1.im !=0 ){
        printf("%.4f%+.4f i and %.4f%+.4f i", r1.re, r1.im, r2.re, r2.im);
    }
    else if(r1.re != 0 && r1.im ==0){
        printf("%.4f and %.4f", r1.re, r2.re);
    }
    else if(r1.re == 0 && r1.im ==0){
        printf("%.4f and %.4f", r1.re, r2.re);
    }
    else{
        printf("%.4f i and %.4f i", r1.im, r2.im);
    }
}
```

Then, using printComplex to print out the complex number in different situation,

```
printComplex(e,f);
verifyxx = time(r2c(a),time(e,e));
verifyx = time(r2c(b),e);
if(absComplex(add(verifyxx,add(verifyx,r2c(c)))) < 0.000001){
    printf("\n\nQuadratic equation solution is a pair of valid roots. Verification succeeds.");
}
```

Verify the result by plugging the root into the Quadratic Equation a*root1^2 + b*root1 + c= 0 and use abs to get the number if < 10^-6 will give the verification succeeds.