ROOTS OF A QUADRATIC EQUATION

CODE

```c
#include <stdio.h>
#include <conio.h>
#include <math.h>
void quadraticroots(float, float, float);
int main(){
    float p,q,r;
    printf("In a Quadratic Equation of form ax^2+bx+c=0, enter the coefficients a, b and c:- \n");
    scanf("%f %f %f", &p, &q, &r);
    quadraticroots(p,q,r);
    getch();
    return 0;
}
void quadraticroots(float a, float b, float c) {
    float discriminant, root1, root2, realPart, imagPart;
    discriminant = pow(b,2) - 4 * a * c;
    // Condition for real and different roots :-
    if (discriminant > 0) {
        root1 = (-b + sqrt(discriminant)) / (2 * a);
        root2 = (-b - sqrt(discriminant)) / (2 * a);
        printf("root1 = %f and root2 = %f", root1, root2);
    }
    // Condition for real and equal roots:-
    else if (discriminant == 0) {
        root1 = root2 = -b / (2 * a);
        printf("root1 = root2 = %f", root1);
    }
    // Condition for non-real roots:-
    else {
        realPart = -b / (2 * a);
        imagPart = sqrt(-discriminant) / (2 * a);
        printf("1st root = (%f + %f i) and 2nd root = (%f - %f i) , where i = (-1)^(0.5)", realPart, imagPart, realPart, imagPart);
    }
    return;
}
```

OUTPUT

```
In a Quadratic Equation of form ax^2+bx+c=0, enter the coefficients a, b and c:-
1 2 3
1st root = (-1.000000 + 1.414214 i) and 2nd root = (-1.000000 - 1.414214 i) , where i = (-1)^(0.5)
```

```
In a Quadratic Equation of form ax^2+bx+c=0, enter the coefficients a, b and c:-
4 5 6
1st root = (-0.625000 + 1.053269 i) and 2nd root = (-0.625000 - 1.053269 i) , where i = (-1)^(0.5)
```

Hand written code

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
void quadricroots(float, float, float);
int main(){
    float p,q,r;
    Printf("In a quadric Equation of form ax^2+bx+c=0, Enter
        the coefficients a,b and c:-\n");
    Scanf("%f %f %f", &p, &q, &r);
    quadratic roots(p,q,r);
    getch();
    return 0;
}
voidquadratic roots(floata, floatb, floatc)
{ float discriminant, root1, root2, realPart, imagPart;
  discriminant = Pow(b,2) - 4*a*c;
```

```
// condition for real and different roots :
if (discriminant > 0)
{ root1 = (-b + sqrt (discriminant)) / (2*a);
  root2 = (-b + sqrt (discriminant)) / (2*a);
  Printf ("root 1 = %.f and root 2 = %.f, root1, root2);
}
// condition for real and Equal roots
  else if (discriminant == 0).
  { root1 = root2 = -b / (2*a);
  Printf ("root1 = %.f and root 2 = %.f", root1, root2);
  Printf ("root 1 = root2 = %.f", root1);
}
// condition for non real roots :
  else { real Part = -b / (2*a);
         imag Part = sqrt (-discriminant) / (2*a);
         Printf ("1st root = (%.f + %.f i) and 2nd root = (%.f - %.f);
         where i = (-1)^(0.5)", real Part, imag Part, real Part,
          imag Part);
  }
  return;
```