

```

1 #include <stdio.h>
2 #include <math.h>
3
4 typedef enum {
5     Sol_Inconsistent = 0,
6     Sol_Linear,
7     Sol_RepeatedRoots,
8     Sol_RealRoots,
9     Sol_ComplexRoots,
10    Sol_Infinite}
11 SolutionType;
12
13 SolutionType Solve(double a, double b, double c, double *r1, double *r2)
14 {
15     SolutionType retVal = Sol_Inconsistent;
16     if (0 == a) {
17         if (0 == b) {
18             if (0 == c) { // Infinite solutions
19                 retVal = Sol_Infinite;
20             } else { // Inconsistent equation
21                 retVal = Sol_Inconsistent;
22             }
23         } else { // Linear equation
24             retVal = Sol_Linear;
25             *r1 = -c/b;
26         }
27     } else {
28         double disc = b*b - 4*a*c;
29         if (0 == disc) { // fabs(disc) < 1.e-8 // Repeated roots
30             retVal = Sol_RepeatedRoots;
31             *r1 = *r2 = -b/(2*a);
32         } else {
33             if (disc > 0) { // Real distinct roots
34                 retVal = Sol_RealRoots;
35                 *r1 = (-b + sqrt(disc))/(2*a);
36                 *r2 = (-b - sqrt(disc))/(2*a);
37             } else { // Complex conjugate roots
38                 retVal = Sol_ComplexRoots;
39                 // ...
40             }
41         }
42     }
43
44     return retVal;
45 }
46
47 int main_qe() {
48     double a, b, c, r1, r2;
49     SolutionType retVal;
50
51     scanf("%lf %lf %lf", &a, &b, &c);
52     retVal = Solve(a, b, c, &r1, &r2);
53     printf("Type of Solution = %d\n", retVal);
54     return 0;
55 }
56

```