#include <iostream>

using namespace std;

| 00 | unsigned int Solve(double a, double b, double c, double& r1, double& r2) |
| --- | --- |
| 01 | { |
| 02 | unsigned int retVal = 0; |
| 03 | if (0 == a) { |
| 04 | if (0 == b) { |
| 05 | if (0 == c) { // Infinite solutions |
| 06 | retVal = 5; |
| 07 | } else { // Inconsistent equation |
| 08 | retVal = 0; |
| 09 | } |
| 10 | } else { // Linear equation |
| 11 | retVal = 1; |
| 12 | r1 = -c/b; |
| 13 | } |
| 14 | } else { |
| 15 | double disc = b\*b - 4\*a\*c; |
| 16 | if (0 == disc) { // Repeated roots |
| 17 | retVal = 2; |
| 18 | r1 = r2 = -b/(2\*a); |
| 19 | } else { |
| 20 | if (disc > 0) { // Real distinct roots |
| 21 | retVal = 3; |
| 22 | r1 = (-b + sqrt(disc))/(2\*a); |
| 23 | r2 = (-b - sqrt(disc))/(2\*a); |
| 24 | } else { // Complex conjugate roots |
| 25 | retVal = 4; |
| 26 | // ... |
| 27 | } |
| 28 | } |
| 29 | } |
| 30 |  |
| 31 | return retVal; |
| 32 | } |

| **a** | **b** | **c** | **Equivalence Class** | **Statements Covered** | **Branches Covered** | **Paths Covered** |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | Infinite roots | 2,3,4,5,6,31 | 2-3,3-4,4-5,5-6,6-31 | 2-3-4-5-6-31 |
| 0 | 0 | 2 | No root | 2,3,4,5,8,31 | 2-3,3-4,4-5,5-8,8-31 | 2-3-4-5-8-31 |
| 0 | 2 | -4 | Single root | 2,3,4,11,12,31 | 2-3,3-4,4-11,11-12,12-31 | 2-3-4-11-12-31 |
| 4 | 4 | 1 | Repeated roots | 2,3,15,16,17,18,31 | 2-3,3-15,15-16,16-17,17-18,18-31 | 2-3-15-16-17-18-31 |
| 1 | -5 | 6 | Distinct roots | 2,3,15,16,20,21,22,23,31 | 2-3,3-15,15-16,16-20,20-21,21-22,22-23,23-31 | 2-3-15-16-20-21-22-23-31 |
| 2 | 3 | 4 | Complex roots | 2,3,15,16,20,25,26,31 | 2-3,3-15,15-16,16-20,20-25,25-26,26-31 | 2-3-15-16-20-25-26-31 |

int main()

{

double a, b, c;

double r1, r2;

unsigned int retVal = 0;

unsigned int moreInputs = 0;

do {

cout << "Input coefficients: a, b and c" << endl;

cin >> a >> b >> c;

cout << endl;

switch (retVal = Solve(a, b, c, r1, r2)) {

case 0: cout << "No root" << endl; break;

case 1: cout << "Linear Eqn: r1 = " << r1 << endl; break;

case 2: cout << "Repeated real roots: r1 = " << r1 << " r2 = " << r2 << endl; break;

case 3: cout << "Distinct real roots: r1 = " << r1 << " r2 = " << r2 << endl; break;

case 4: cout << "Complex conjugate roots" << endl; break;

case 5: cout << "Infinite roots" << endl; break;

default: cout << "Something wrong" << endl; break;

}

cout << "Continue Solving? Input 1" << endl;

cin >> moreInputs;

} while (1 == moreInputs);

return 0;

}