```cpp
#include <iostream>
using namespace std;

unsigned int Solve(double a, double b, double c, double& r1, double& r2)
{
	unsigned int retVal = 0;
	if (0 == a) {
		if (0 == b) {
			if (0 == c) { // Infinite solutions
				retVal = 5;
			} else { // Inconsistent equation
				retVal = 0;
			}
		} else { // Linear equation
			retVal = 1;
			r1 = -c/b;
		}
	} else {
		double disc = b*b - 4*a*c;
		if (0 == disc) { // Repeated roots
			retVal = 2;
			r1 = r2 = -b/(2*a);
		} else {
			if (disc > 0) { // Real distinct roots
				retVal = 3;
				r1 = (-b + sqrt(disc))/(2*a);
				r2 = (-b - sqrt(disc))/(2*a);
			} else { // Complex conjugate roots
				retVal = 4;
				// ...
			}
		}
	}

	return retVal;
}
```

Test Cases:

a	b	c	Case
0	0	0	Infinite roots
0	0	2	No root
0	2	-4	Single root
4	4	1	Repeated roots
1	-5	6	Distinct roots
2	3	4	Complex roots

```
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;
}
```