```
using System;
class Program
{
	static void Main()
	{
		double a, b, c;

		(a, b, c) = GetCoefficients();

		double[] result = CalculateRoots(a, b, c);

		if (result == null)
		{
			Console.WriteLine("Корней нет");
		}
		else
		{
			foreach (var root in result)
			{
				Console.WriteLine(root);
			}
		}
	}

	static (double, double, double) GetCoefficients()
	{
		try
		{
			Console.WriteLine("Введите коэффициенты a, b и c");
			string[] input = Console.ReadLine().Split();
```

```
            double a = double.Parse(input[0]);
            double b = double.Parse(input[1]);
            double c = double.Parse(input[2]);

            return (a, b, c);
        }
        catch
        {
            Console.WriteLine("Ошибка ввода данных");
            return GetCoefficients();
        }
}

static double[] CalculateRoots(double a, double b, double c)
{
        double D = b * b - 4 * a * c;

        if (D < 0)
        {
            return null;
        }
        else if (D > 0)
        {
            double[] roots = new double[4];
            try
            {
                double r2_1 = (-b + Math.Sqrt(D)) / (2 * a);
                double r2_2 = (-b - Math.Sqrt(D)) / (2 * a);

                if (r2_1 >= 0)
```

```
                {
                    double x1 = Math.Sqrt(r2_1);
                    double x2 = -Math.Sqrt(r2_1);
                    roots[0] = x1;
                    roots[1] = x2;
                }

                if (r2_2 >= 0)
                {
                    double x3 = Math.Sqrt(r2_2);
                    double x4 = -Math.Sqrt(r2_2);
                    roots[2] = x3;
                    roots[3] = x4;
                }
            }
            catch
            {
                Console.WriteLine("Не биквадратное
уравнение");
                Environment.Exit(1);
            }

            if (roots[0] == 0 && roots[2] == 0)
            {
                return null;
            }

            return roots;
        }
        else
```

```
        {
            double t = -b / (2 * a);

            if (t > 0)
            {
                double[] roots = new double[2];
                double x1 = Math.Sqrt(t);
                double x2 = -Math.Sqrt(t);
                roots[0] = x1;
                roots[1] = x2;
                return roots;
            }
            else
            {
                return null;
            }
        }
    }
}
```