

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ  
Н. Э. БАУМАНА

РАДИОТЕХНИЧЕСКИЙ ФАКУЛЬТЕТ

КАФЕДРА «СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ»

**ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1**  
**курса «Базовые компоненты интернет-технологий»**  
**«ОСНОВЫ ЯЗЫКА C<sup>#</sup>»**

Выполнил студент:

Елисеев Глеб Борисович

группа: РТ5-31

Проверил:

к.т.н., доцент

Гапанюк Юрий Евгеньевич

Москва, 2017 г.

**Описание задания.** Разработать программу для решения квадратного уравнения:

1. Программа должна быть разработана в виде консольного приложения на языке C#.
2. Программа осуществляет ввод с клавиатуры коэффициентов A, B, C, вычисляет дискриминант и корни уравнения (в зависимости от дискриминанта).
3. Если коэффициент A, B, C введен некорректно, то необходимо проигнорировать некорректное значение и ввести коэффициент повторно.

#### Исходный код 1 — Program.cs

```
using System;

namespace lab1
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Введите коэффициенты A, B и C квадратного
                               ↪ уравнения.");

            var a = ReadDouble();
            while (a == 0)
            {
                Console.WriteLine("Коэффициент A не может быть равным 0. Введите
                                   ↪ число повторно!");
                a = ReadDouble();
            }
            var b = ReadDouble();
            var c = ReadDouble();

            double[] roots = SolveEquation(a, b, c);
```

```

        PrintRoots(roots);
        Console.ReadKey();
    }

    /// <summary>
    /// Считывает строковое представление числа из консоли и пытается
    ///     ↪ преобразовать его в эквивалентное число
    /// двойной точности с плавающей запятой.
    /// </summary>
    /// <returns>Число двойной точности с плавающей запятой</returns>
    static double ReadDouble()
    {
        double d = 0.0D;
        bool isParsable = false;

        Console.Write("Введите число: ");
        while (!isParsable)
        {
            isParsable = double.TryParse(Console.ReadLine(), out d);
            if (!isParsable)
                Console.Write("Некорректное значение. Введите число
                    ↪ повторно: ");
        }

        return d;
    }

    /// <summary>
    /// Вычисляет корни квадратного уравнения (в зависимости от
    ///     ↪ дискриминанта).
    /// </summary>
    /// <param name="a">Старший коэффициент уравнения</param>
    /// <param name="b">Средний коэффициент уравнения</param>
    /// <param name="c">Свободный член уравнения</param>
    /// <returns>Корни квадратного уравнения (числа двойной точности с
    ///     ↪ плавающей запятой)</returns>
    static double[] SolveEquation(double a, double b, double c)
    {
        double[] roots;
        double d = Math.Sqrt(CountDiscriminant(a, b, c));
    }

```

```

    if (double.IsNaN(d))
        roots = new double[0];
    else
        roots = new double[2] { CountRoot(a, b, d), CountRoot(a, b, -d)
        ↪ };

    return roots;
}

/// <summary>
/// Вычисляет дискриминант квадратного уравнения.
/// </summary>
/// <param name="a">Старший коэффициент уравнения</param>
/// <param name="b">Средний коэффициент уравнения</param>
/// <param name="c">Свободный член уравнения</param>
/// <returns>Дискриминант квадратного уравнения (число двойной точности
    ↪ с плавающей запятой)</returns>
static double CountDiscriminant(double a, double b, double c) =>
    ↪ Math.Pow(b, 2) - 4 * a * c;

/// <summary>
/// Вычисляет корень квадратного уравнения (в зависимости от квадратного
    ↪ корня из дискриминанта).
/// </summary>
/// <param name="a">Старший коэффициент уравнения</param>
/// <param name="b">Средний коэффициент уравнения</param>
/// <param name="sqrt_discriminant">Квадратный корень из
    ↪ дискриминанта</param>
/// <returns>Корень квадратного уравнения (число двойной точности с
    ↪ плавающей запятой)</returns>
static double CountRoot(double a, double b, double sqrt_discriminant) =>
    ↪ (sqrt_discriminant - b) / (2 * a);

/// <summary>
/// Если длина массива равна 2, то выводит в консоль корни квадратного
    ↪ уравнения.
/// Если же длина массива равна 0, то выводит сообщение о том, что
    ↪ данное квадратное
/// уравнение не имеет корней.
/// </summary>
/// <param name="roots">Корни квадратного уравнения</param>

```

```
static void PrintRoots(double[] roots)
{
    if (roots.Length == 2)
        Console.WriteLine($"Корни данного квадратного уравнения:
        ↪ {roots[0]}, {roots[1]}.");
    else if (roots.Length == 0)
        Console.WriteLine(@"Данное квадратное уравнение не имеет
        ↪ корней.");
    }
}
```

---

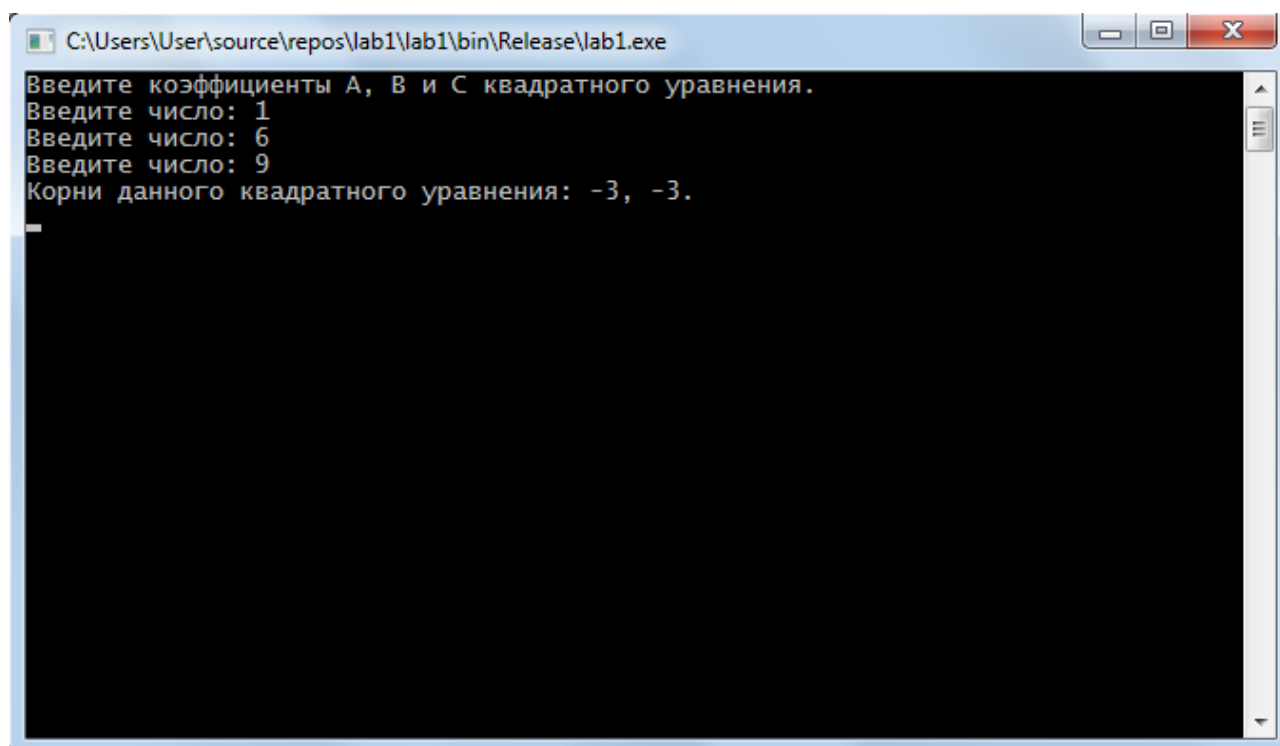


Рисунок 1 — Значение дискриминанта квадратного уравнения  $D = 0$ .

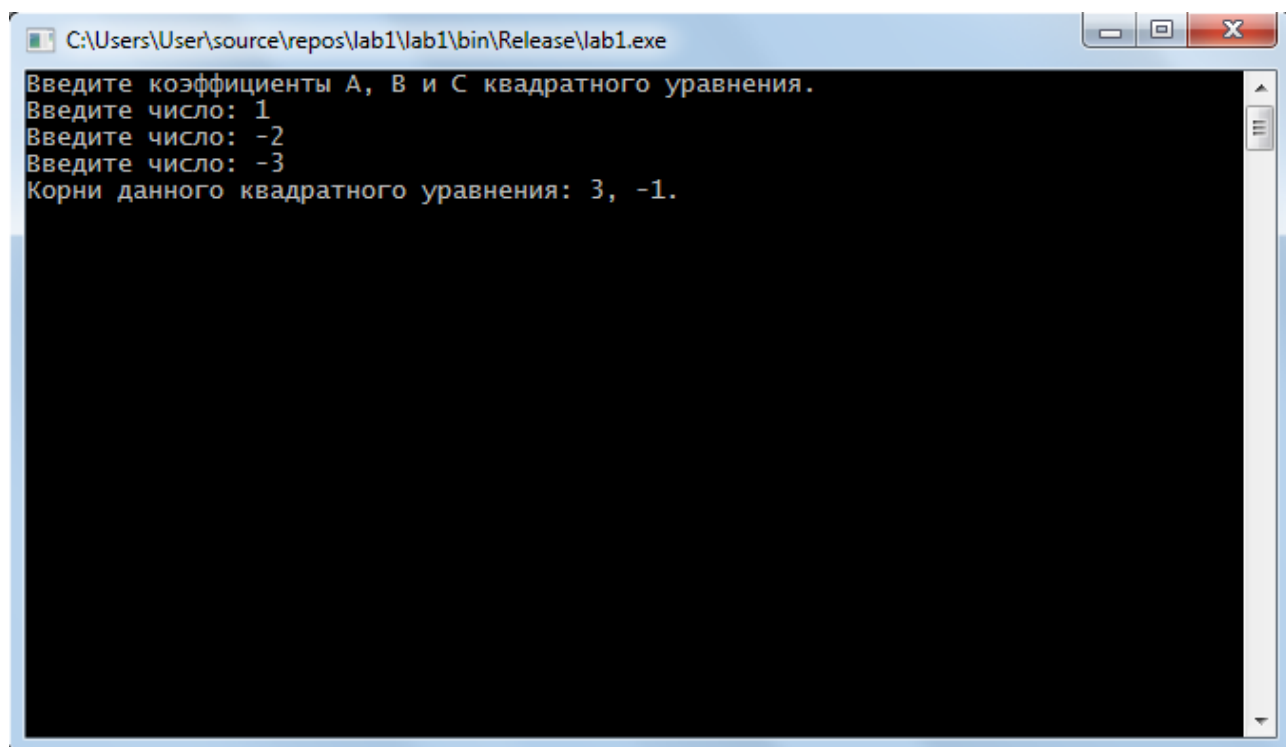


Рисунок 2 — Значение дискриминанта квадратного уравнения  $D > 0$ .

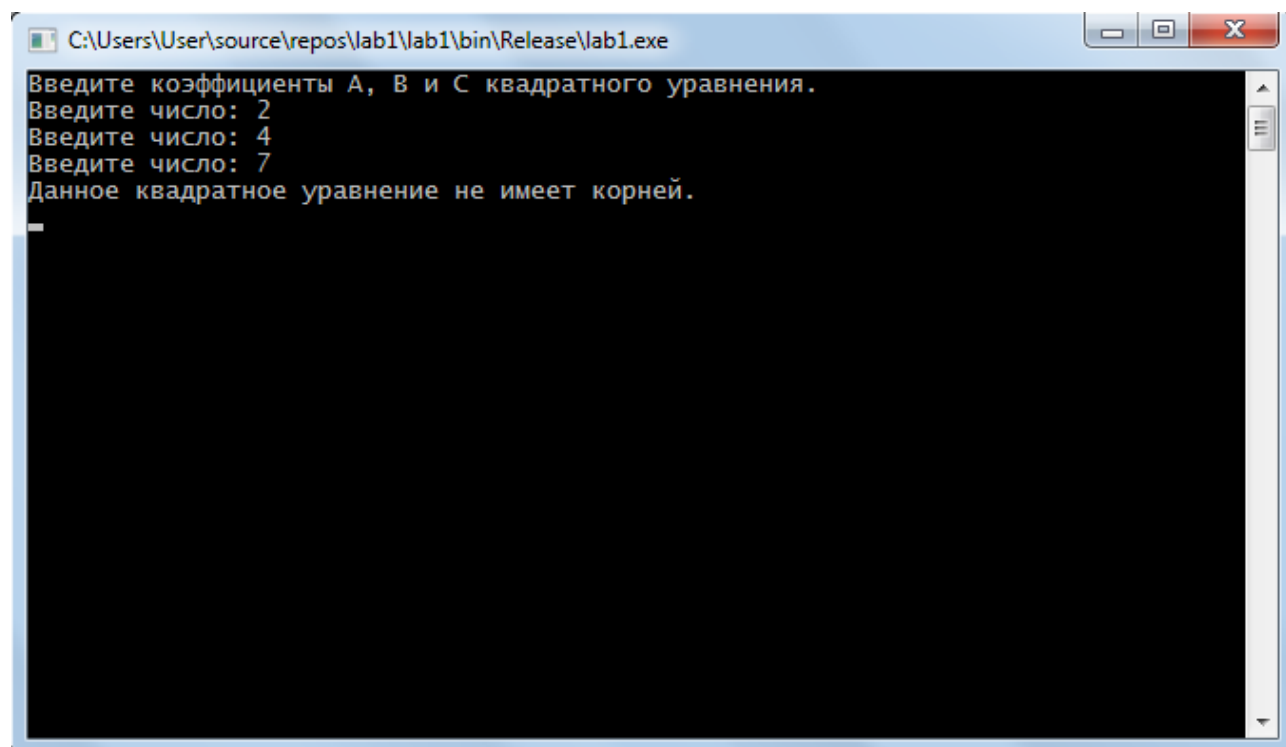


Рисунок 3 — Значение дискриминанта квадратного уравнения  $D < 0$ .

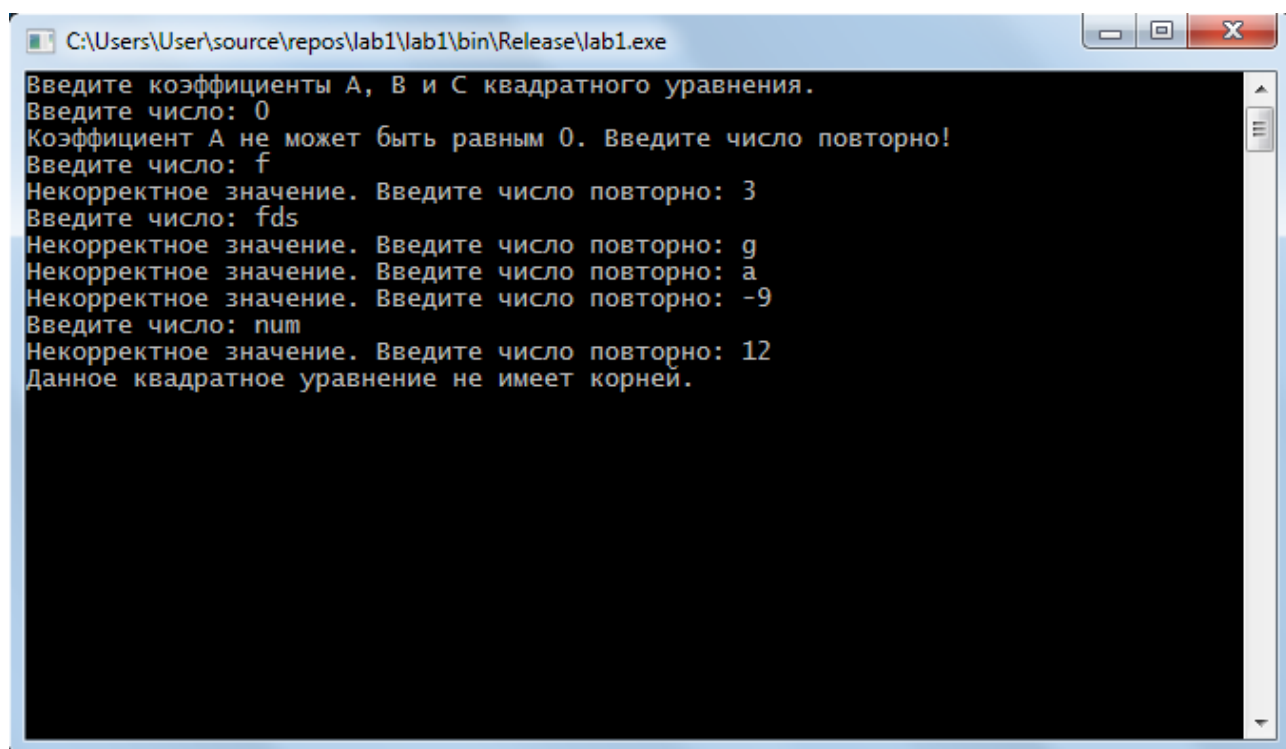


Рисунок 4 — Некорректные значения коэффициентов А, В и С.