**Задание 1**

программа на C#, которая вычисляет значения функции на заданном интервале от

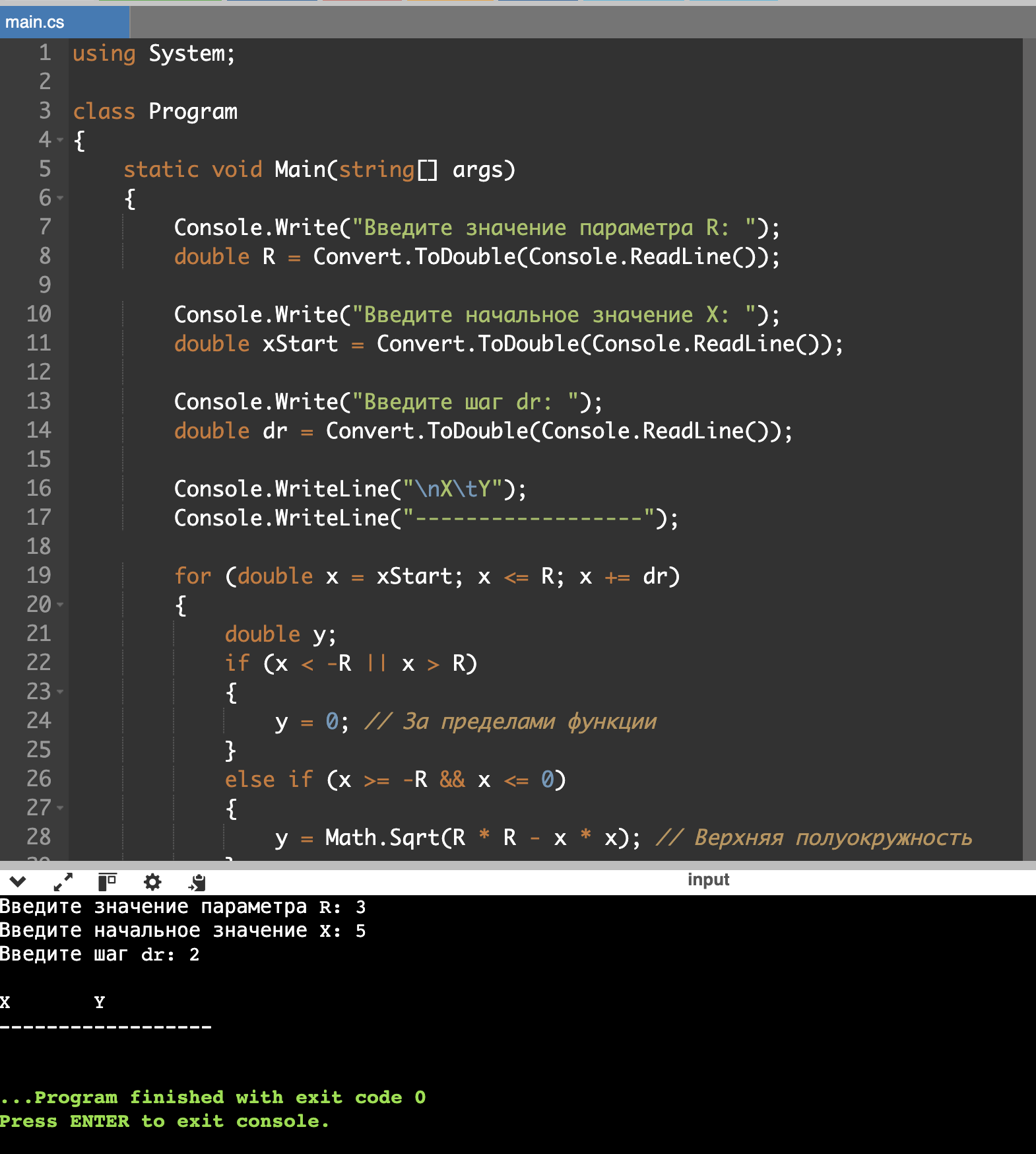
x

*x* до

x+dr

*x*+*dr* и выводит результаты в виде таблицы. Проигнорированы ветвления, а вместо этого используется единая формула, аналогичная предыдущему примеру:

Результат работы программы -



using System;

class Program  
 {  
 static void Main(string[] args)  
 {  
 Console.Write("Введите значение параметра R: ");  
 double R = Convert.ToDouble(Console.ReadLine());

Console.Write("Введите начальное значение X: ");

double xStart = Convert.ToDouble(Console.ReadLine());

Console.Write("Введите шаг dr: ");

double dr = Convert.ToDouble(Console.ReadLine());

Console.WriteLine("\nX\tY");

Console.WriteLine("------------------");

for (double x = xStart; x <= R; x += dr)

{

double y;

if (x < -R || x > R)

{

y = 0; // За пределами функции

}

else if (x >= -R && x <= 0)

{

y = Math.Sqrt(R \* R - x \* x); // Верхняя полуокружность

}

else

{

y = Math.Sqrt(R \* R - x \* x); // Верхняя полуокружность

}

Console.WriteLine($"{x:F2}\t{y:F2}");

}

}

}

**ЗАДАНИЕ 2**

Для попадания мишени нужно вводить координаты для кажого выстрела после чего программа определяет попадает ли каждая точка в заштрихоанную область .  
  
using System;

class Program  
 {  
 static void Main(string[] args)  
 {  
 Console.Write("Введите радиус R: ");  
 double R = Convert.ToDouble(Console.ReadLine());

for (int i = 1; i <= 10; i++)

{

Console.Write($"Введите координаты точки {i} (x, y): ");

string[] input = Console.ReadLine().Split(',');

double x = Convert.ToDouble(input[0].Trim());

double y = Convert.ToDouble(input[1].Trim());

// Проверка попадания в заштрихованную область

if (x >= 0 && y >= 0 && x <= R && y <= R)

{

// Проверка попадания под прямую y = x

if (y <= x && (x \* x + y \* y) <= (R \* R))

{

Console.WriteLine($"Выстрел {i}: Точка ({x}, {y}) попадает в заштрихованную область.");

}

else

{

Console.WriteLine($"Выстрел {i}: Точка ({x}, {y}) не попадает в заштрихованную область.");

}

}

else

{

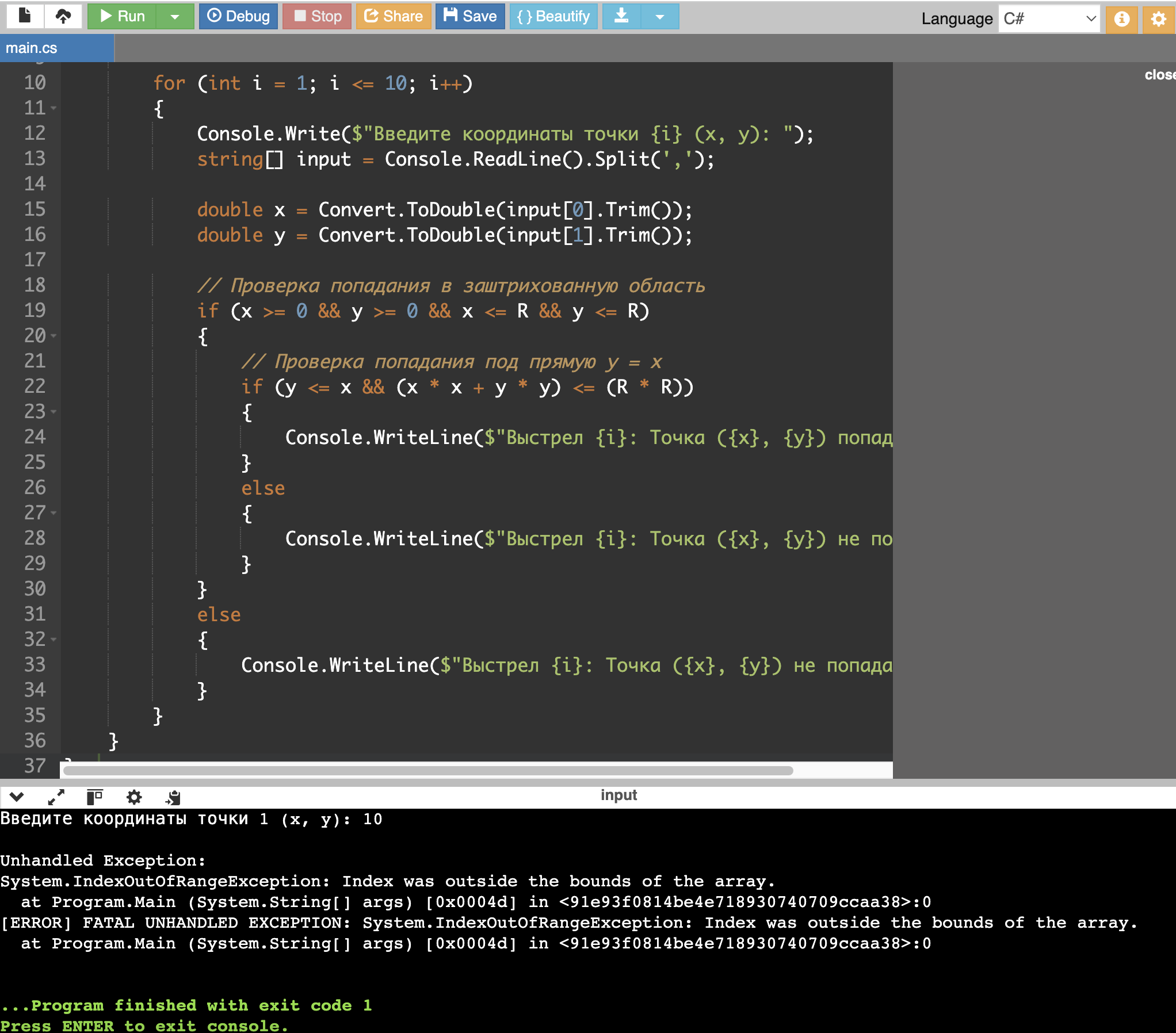
Console.WriteLine($"Выстрел {i}: Точка ({x}, {y}) не попадает в заштрихованную область.");

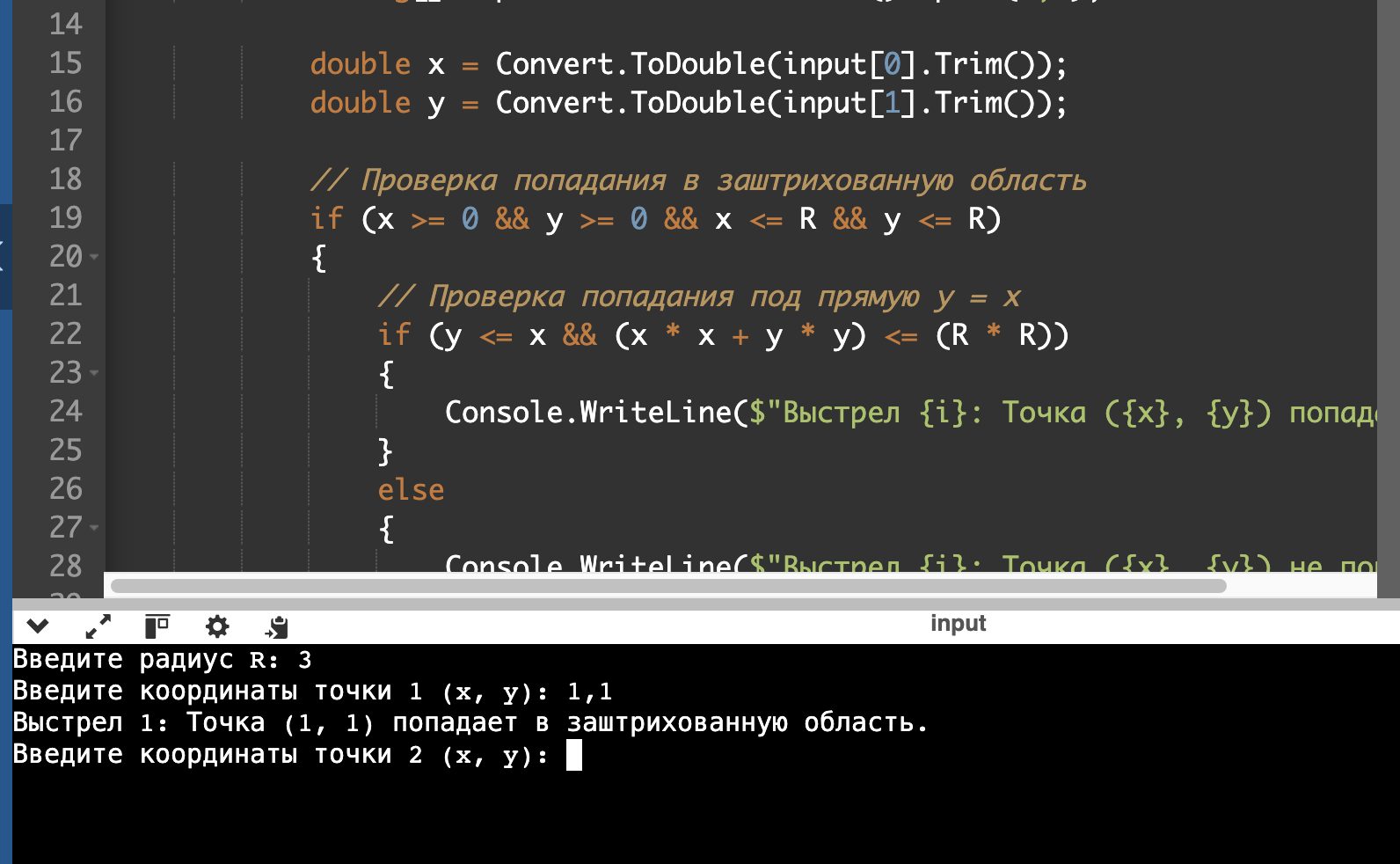
}

}

}

}





**Задание 3**

**using System;**

**class Program  
 {  
 static void Main(string[] args)  
 {  
 Console.Write("Введите начальное значение x: ");  
 double xStart = Convert.ToDouble(Console.ReadLine());**

**Console.Write("Введите конечное значение x: ");**

**double xEnd = Convert.ToDouble(Console.ReadLine());**

**Console.Write("Введите шаг dx: ");**

**double dx = Convert.ToDouble(Console.ReadLine());**

**Console.Write("Введите количество членов ряда для суммирования: ");**

**int terms = Convert.ToInt32(Console.ReadLine());**

**Console.WriteLine("\nX\tValue\tTerms");**

**Console.WriteLine("-------------------------");**

**// Вычисление значений функции для заданного интервала**

**for (double x = xStart; x <= xEnd; x += dx)**

**{**

**double value = CalculateTaylorSeries(x, terms);**

**Console.WriteLine($"{x:F2}\t{value:F6}\t{terms}");**

**}**

**}**

**static double CalculateTaylorSeries(double x, int terms)**

**{**

**// Проверка, что x > 1, так как функция определена только для этого значения**

**if (x <= 1)**

**{**

**throw new ArgumentException("x должно быть больше 1.");**

**}**

**double result = 0.0;**

**for (int n = 0; n < terms; n++)**

**{**

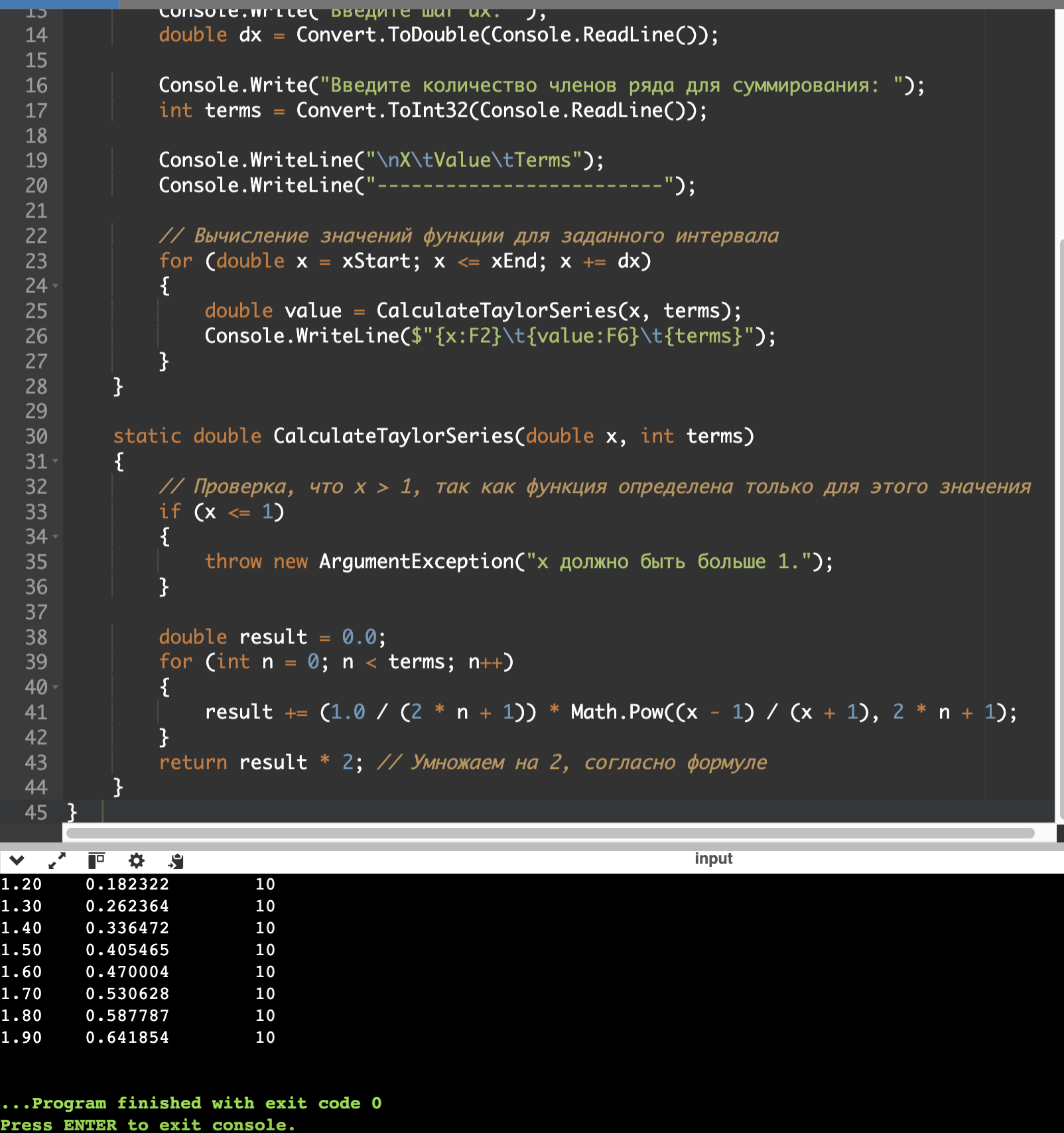
**result += (1.0 / (2 \* n + 1)) \* Math.Pow((x - 1) / (x + 1), 2 \* n + 1);**

**}**

**return result \* 2; // Умножаем на 2, согласно формуле**

**}**

**}**

****