**ПРАКТИЧЕСКОЕ ЗАДАНИЕ 1**

**Принципы объектно -ориентированного программирования**

Задание 1. Дана масса в килограммах. Найти число полных тонн в ней.

Листинг программы:

{

static void Main(string[] args)

{

Console.Write("Ведите число килограмм: ");

int cm, a, b;

cm = Convert.ToInt32(Console.ReadLine());

b = (cm - (cm % 1000)) / 1000;

a = cm % 100;

Console.WriteLine($"Результат: {b} тонны, {a} килограмм");

Console.ReadKey();

}

}

}

Таблица 1.1 – Входные и выходные данные

|  |  |  |
| --- | --- | --- |
| Входные данные | Выходные данные | |
| 3250 кг | 3 тонны | 250 килограмм |

Анализ результатов:

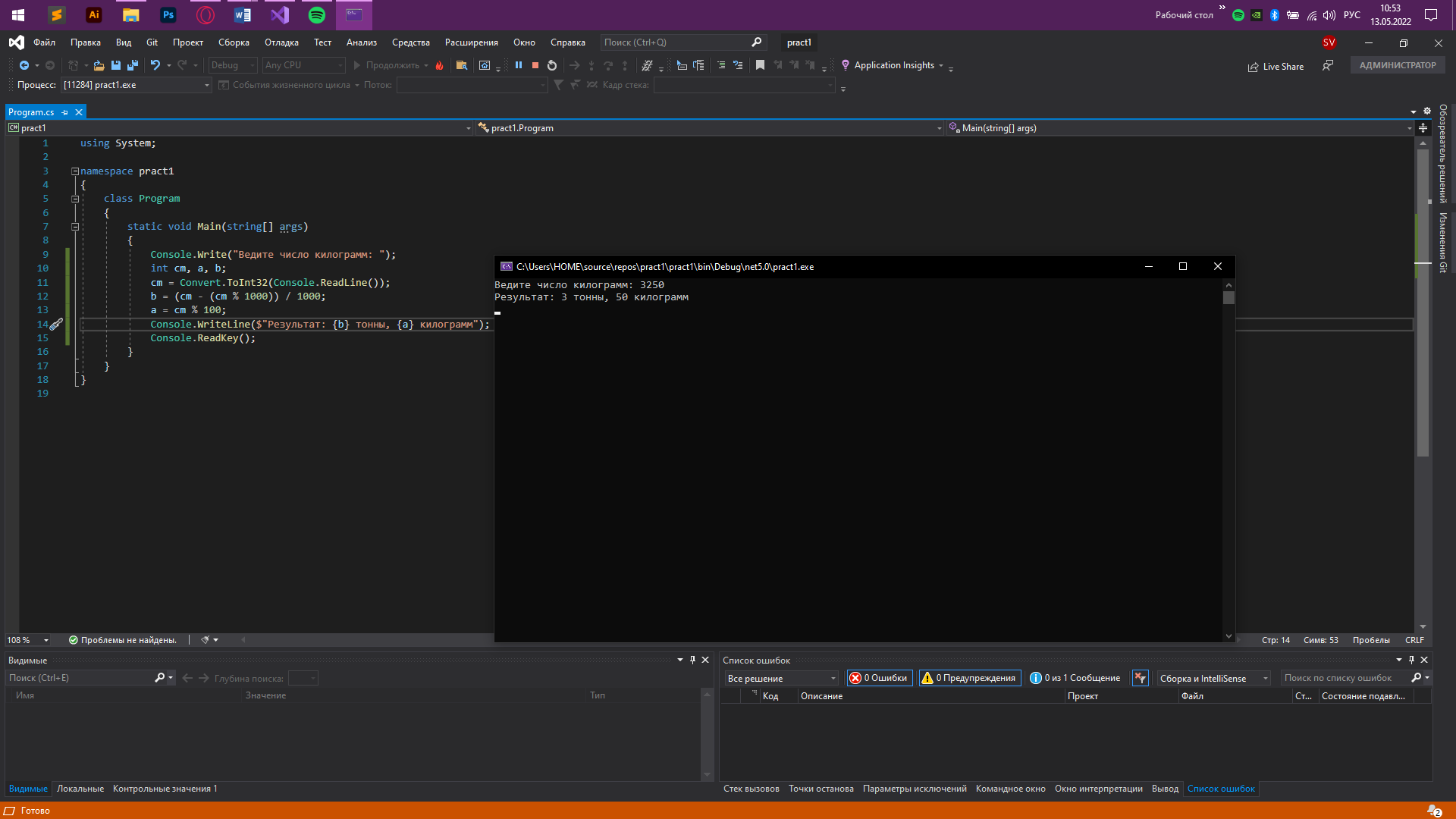


Рисунок 1.1 – Результат работы программы

Источник: собственная разработка

Задание 2. Проверить истинность высказывания: "Данные числа x, y являются координатами точки, лежащей в первой или третьей координатной четверти".

Листинг программы:

{

public static void Main(string[] args)

{

if (double.TryParse(Console.ReadLine(), out var x) && double.TryParse(Console.ReadLine(), out var y))

{

if (x > 0 && y > 0)

Console.WriteLine("x и y в первой четверти");

else if (x < 0 && y < 0)

Console.WriteLine("x и y в третьей четверти");

else

Console.WriteLine("x и y где-то в другом месте");

}

}

}

Таблица 1.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 5 | x и y в первой четверти |
| 10 |

Анализ результатов:

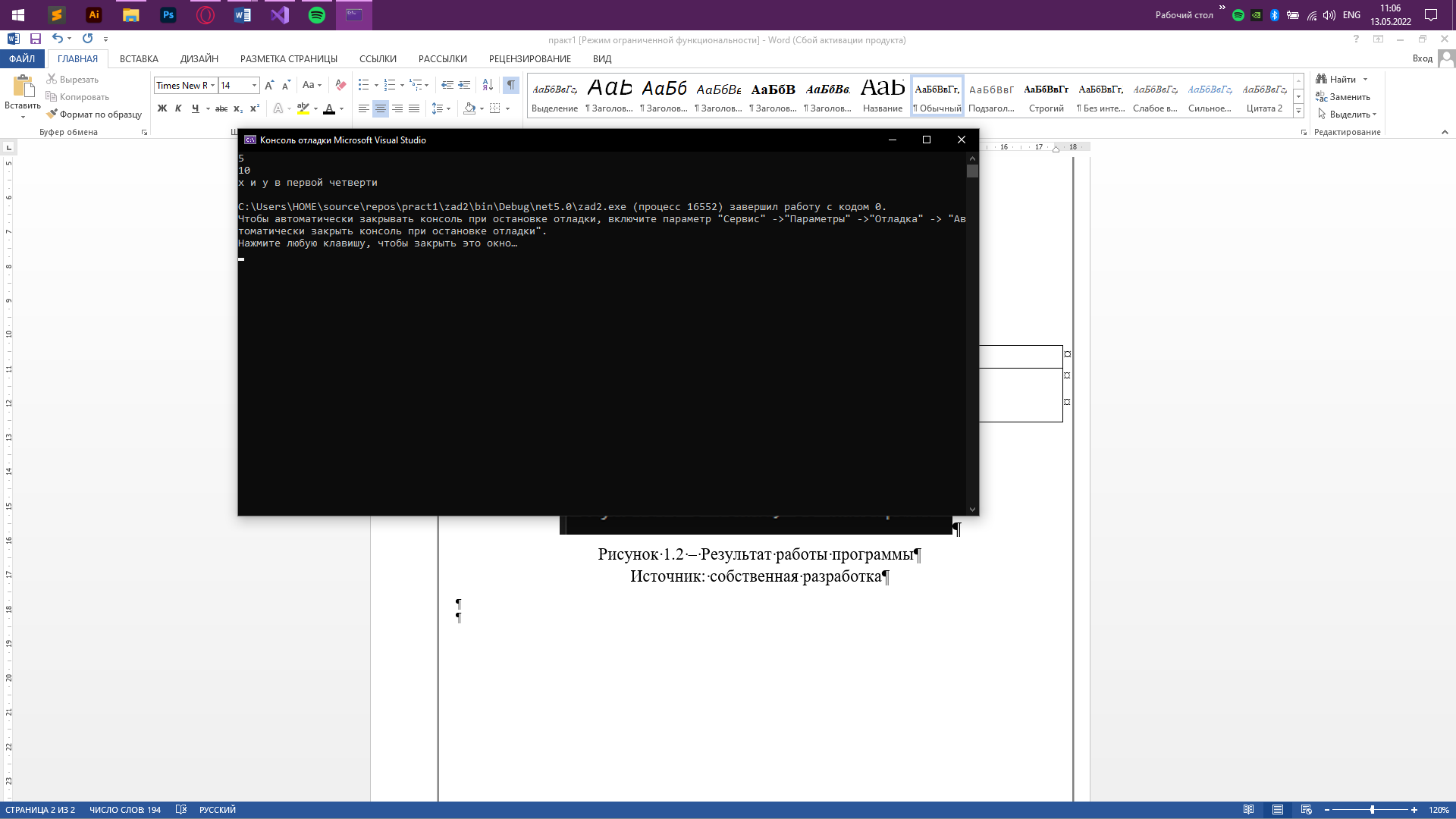


Рисунок 1.2 – Результат работы программы

Источник: собственная разработка

Задание 3. Даны два целых числа A и B (A < B). Вывести в порядке убывания все целые числа, расположенные между A и B (не включая числа A и B), а также количество N этих чисел.

Листинг программы:

{

static void Main(string[] args)

{

Console.Write("a=");

int a = Convert.ToInt32(Console.ReadLine());

Console.Write("b=");

int b = Convert.ToInt32(Console.ReadLine());

for (int i = b - 1; i > a; i--)

Console.WriteLine(i);

Console.WriteLine(b - a - 1);

Console.ReadKey();

}

}

}

Таблица 1.3 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| два целых числа A,B (1<=A,B<=100) | вывести в порядке убывания все целые числа, расположенные между A и B (не включая числа A и B), а также количество N этих чисел |

Анализ результатов:

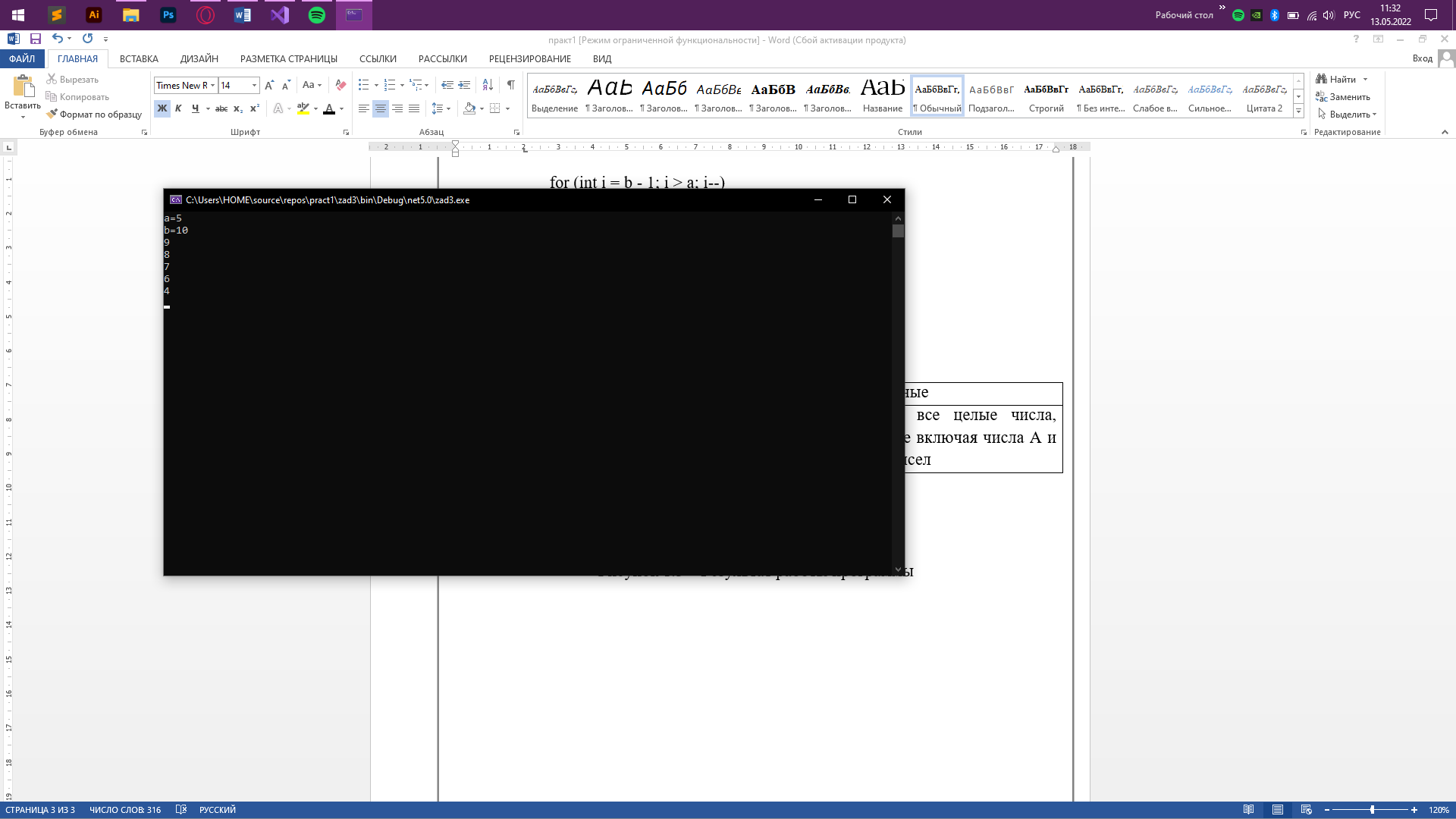


Рисунок 1.3 – Результат работы программы

Источник: собственная разработка

Задание 4. Написать программу, которая вычисляет значение функции y:



Листинг программы:

static void Main(string[] args)

{

const double E = 2.71;

double X;

double Y = 0;

Console.Write("Input Х: ");

X = double.Parse(Console.ReadLine());

if ((X >= 1) && (X <= 2))

{

Y = Math.Sqrt(X) \* Math.Log10(X) \* 10;

}

else

if (X < 1)

{

Y = 1;

}

else

if (X > 2)

{

Y = Math.Pow(E, 2 \* X) \* Math.Cos(5 \* X);

}

Console.WriteLine("Answer: " + Y);

Console.ReadLine();

}

}

Таблица 1.4 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| X = 2 | 3 |

Анализ результатов:

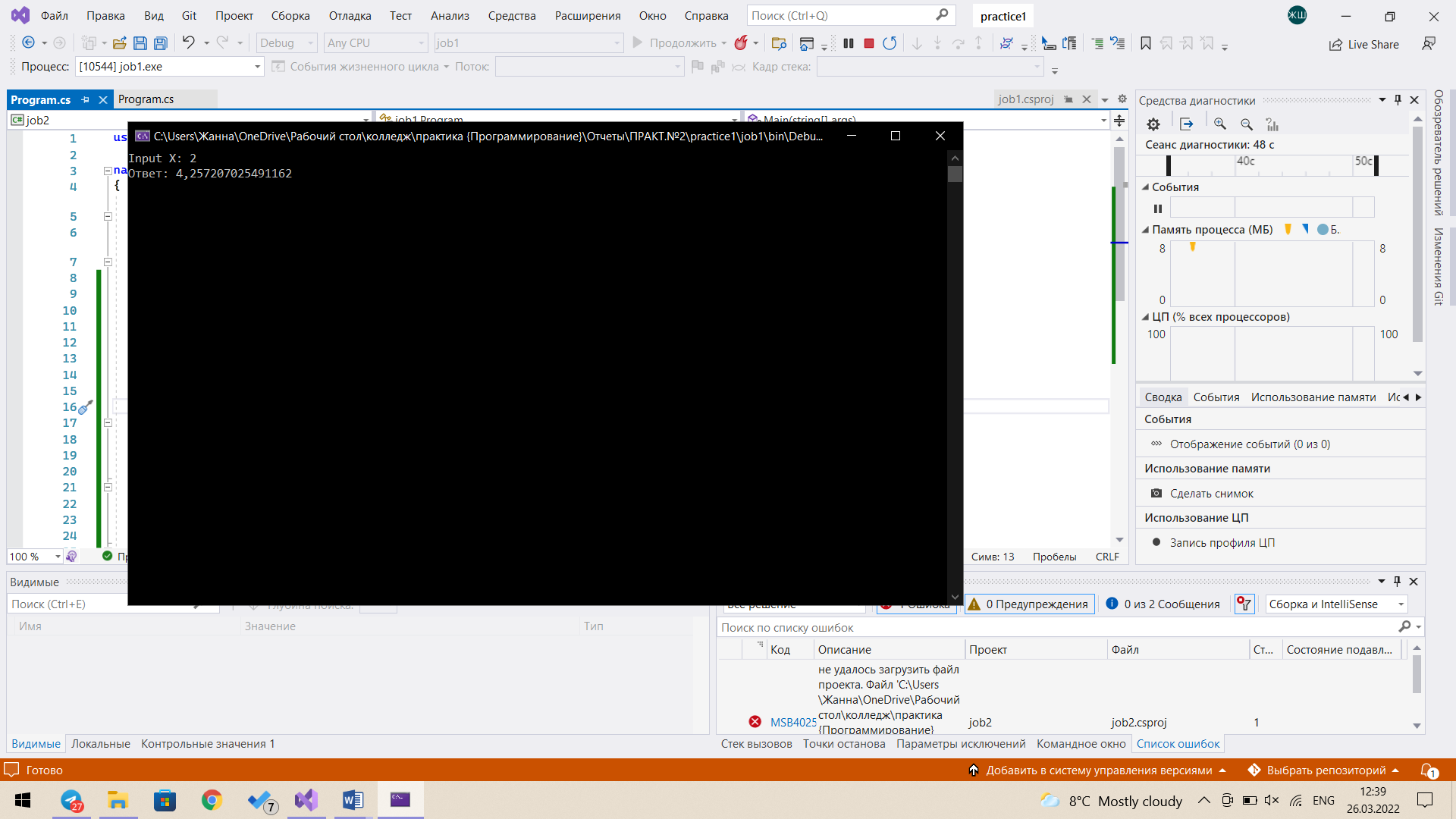


Рисунок 1.4 – Результат работы программы

Источник: собственная разработка

Задание 5. Написать программу, которая определяет является ли заданное целое число нечетным.

Листинг программы:

{

static void Main(string[] args)

{

int number = Convert.ToInt32(Console.ReadLine());

if (number%2 == 0)

{

Console.WriteLine("Число четное");

}

else

{

Console.WriteLine("Число нечетное");

}

}

}

}

Таблица 1.5 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 10 | Число четное |

Анализ результатов:

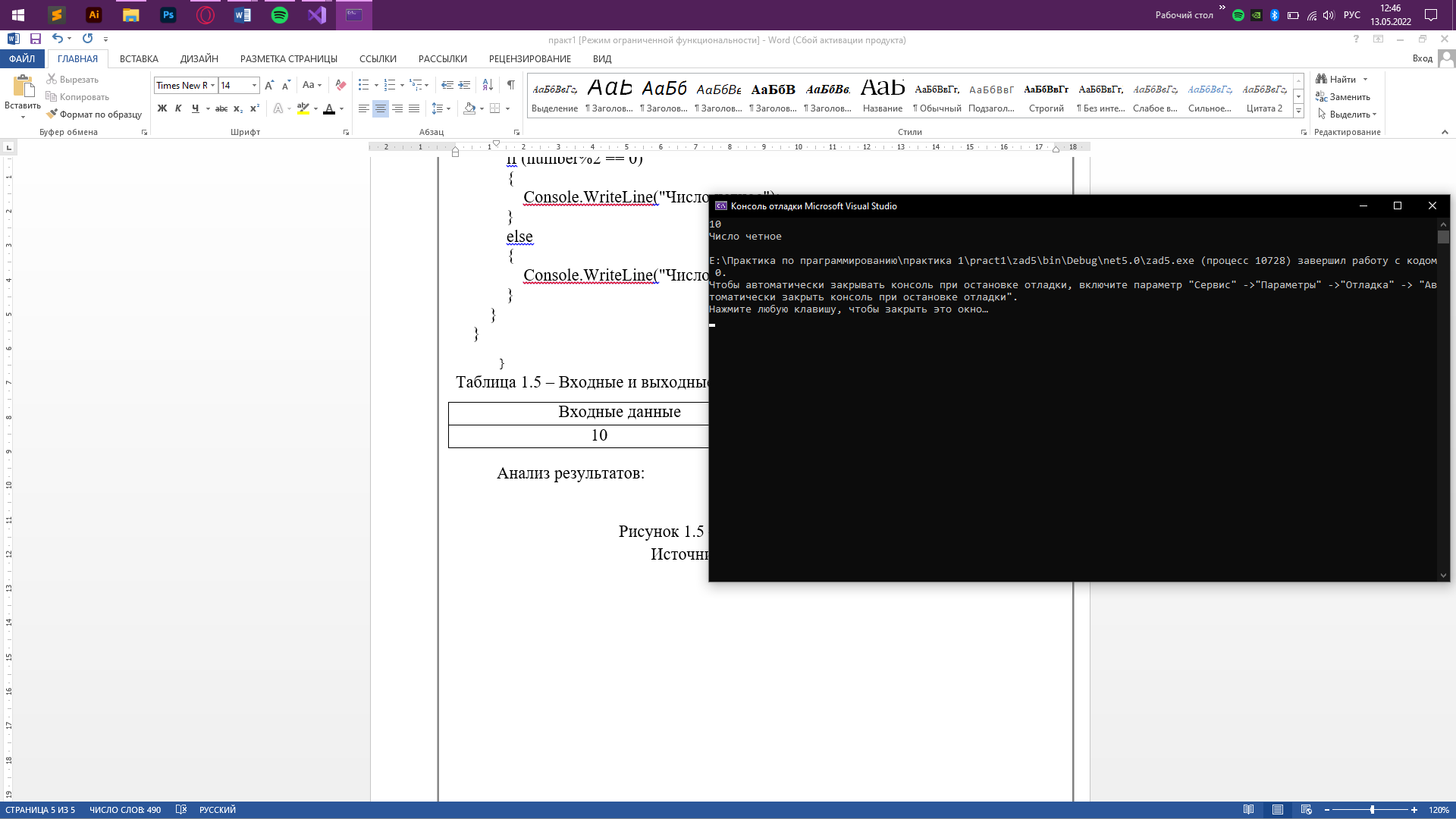


Рисунок 1.5 – Результат работы программы

Источник: собственная разработка

Задание 6. Написать программу, которая выводит на экран возрастную категорию: до года - "младенец", от года до 11 лет - "ребенок", от 12 до 15 лет - "подросток", от 16 до 25 лет - "юноша", от 26 до 70 лет - "мужчина", более c лет - "старик".

Листинг программы:

static void Main(string[] args)

{

Console.Write("n = ");

if (int.TryParse(Console.ReadLine(), out var n))

{

Console.WriteLine(GetCategory(n));

}

else Console.WriteLine("Вы ошиблись");

Console.ReadKey();

string GetCategory(int age)

{

if (age < 1) return "Младенец";

if (1 <= age && age <= 11) return "Ребёнок";

if (12 <= age && age <= 15) return "Подросток";

if (16 <= age && age <= 25) return "Юноша";

if (26 <= age && age <= 70) return "Мужчина";

return "Пожилой человек";

}

}

}

}

}

Таблица 1.6 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 71 | Пожилой человек |

Анализ результатов:

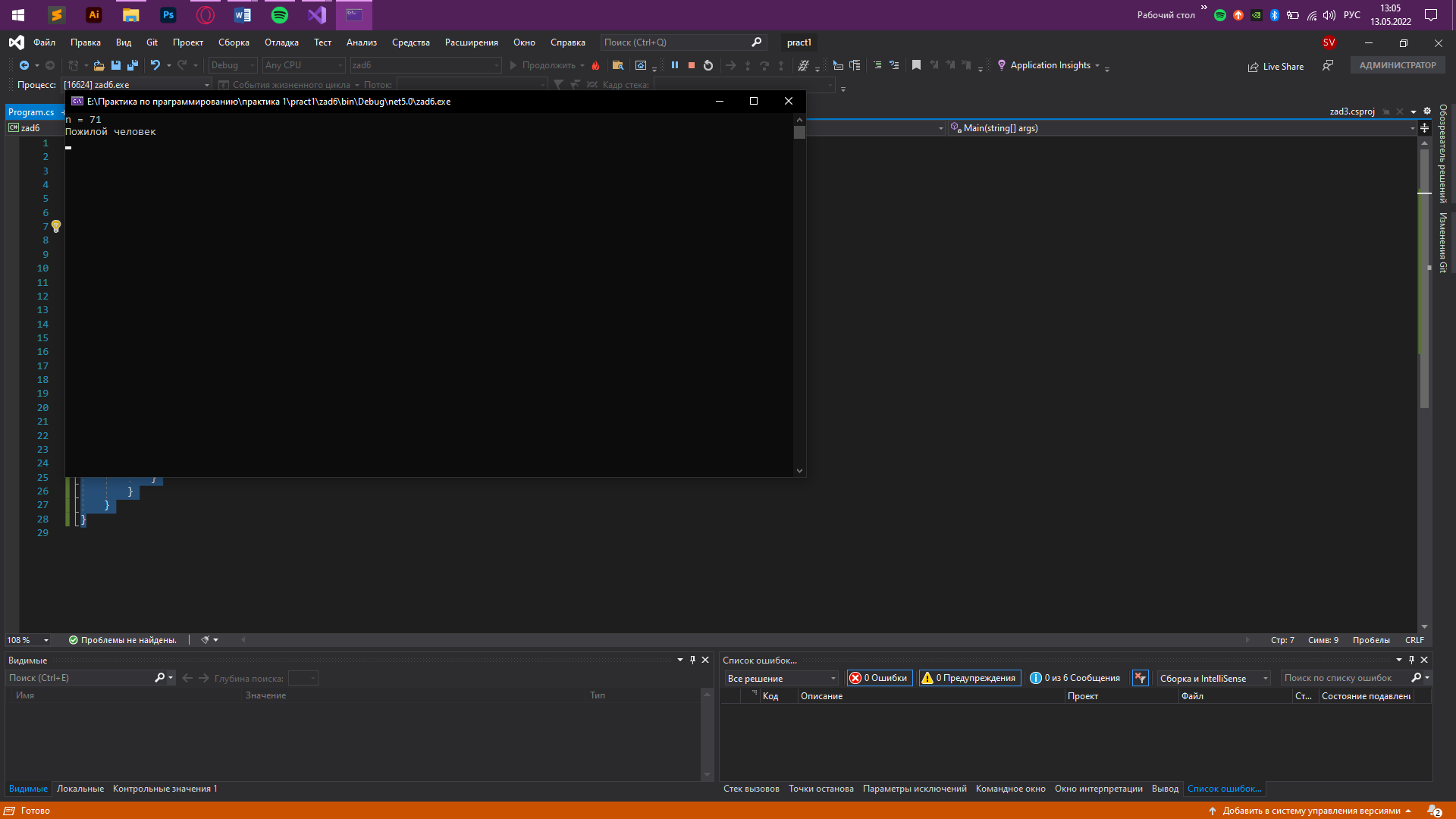


Рисунок 1.6 – Результат работы программы

Источник: собственная разработка

Задание 7. Вывести на экран (задачу решите тремя способами - используя операторы цикла while, do while и for): таблицу соответствия между весом в фунтах и весом в килограммах для значений 1, 2, 3, …, 100 фунтов (1 фунтов = 453г).

Листинг программы:

1) static void Main(string[] args)

{

int i = 1;

while (i <= 100)

{

Console.WriteLine($" { i } фт = { i \* 0.453:N3} кг");

i++;

}

}

}

}

2) static void Main(string[] args)

{

int i = 1;

do

{

Console.WriteLine($" { i } фт = { i \* 0.453:N3} кг");

i++;

}

while (i <= 100);

}

}

}

3) static void Main(string[] args)

{

for (int i = 1; i <= 100; i++)

{

Console.WriteLine($" { i } фт = { i \* 0.453:N3} кг");

}

}

}

}

Таблица 1.7 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 1 | 1 фт = 0,453 кг  2 фт = 0,906 кг  3 фт = 1,359 кг  4 фт = 1,812 кг  5 фт = 2,265 кг  6 фт = 2,718 кг  7 фт = 3,171 кг  8 фт = 3,624 кг  9 фт = 4,077 кг |

Анализ результатов:

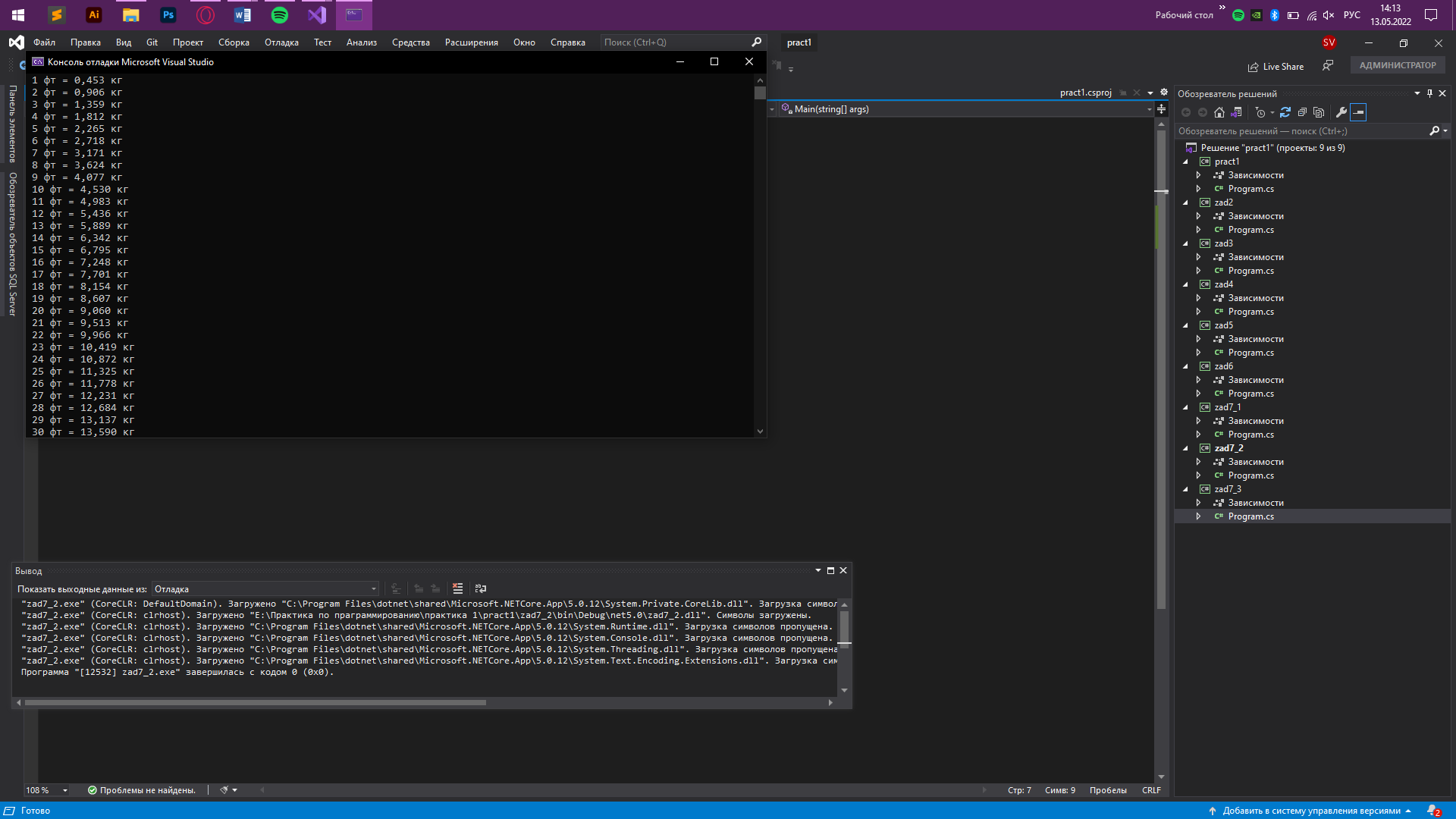


Рисунок 1.7 – Результат работы программы

Источник: собственная разработка

Задание 8. Дано целое число N (N>0). Найти квадрат данного числа, используя для его вычисления следующую формулу: N 2 = 1 + 3 + 5 + … + (2·N – 1). После добавления к сумме каждого слагаемого выводить текущее значение суммы (в результате будут выведены квадраты всех целых чисел от 1 до N).

Входные данные: ввести целое число N (1<=n<=10).

Выходные данные: вывести текущее значение суммы после добавления к сумме каждого слагаемого.

Листинг программы:

static int Sqr(int val)

{

int result = 0;

for (int i = 1; i <= 2 \* val - 1; i = i + 2)

result += i;

return result;

}

static void Main(string[] args)

{

int val = 9;

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

Console.WriteLine(i + ": " + Sqr(i));

Console.ReadKey(true);

}

}

Таблица 1.8 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| N (1<=n<=10). | 1: 1  2: 4  3: 9  4: 16  5: 25  6: 36  7: 49  8: 64  9: 81 |

Анализ результатов:

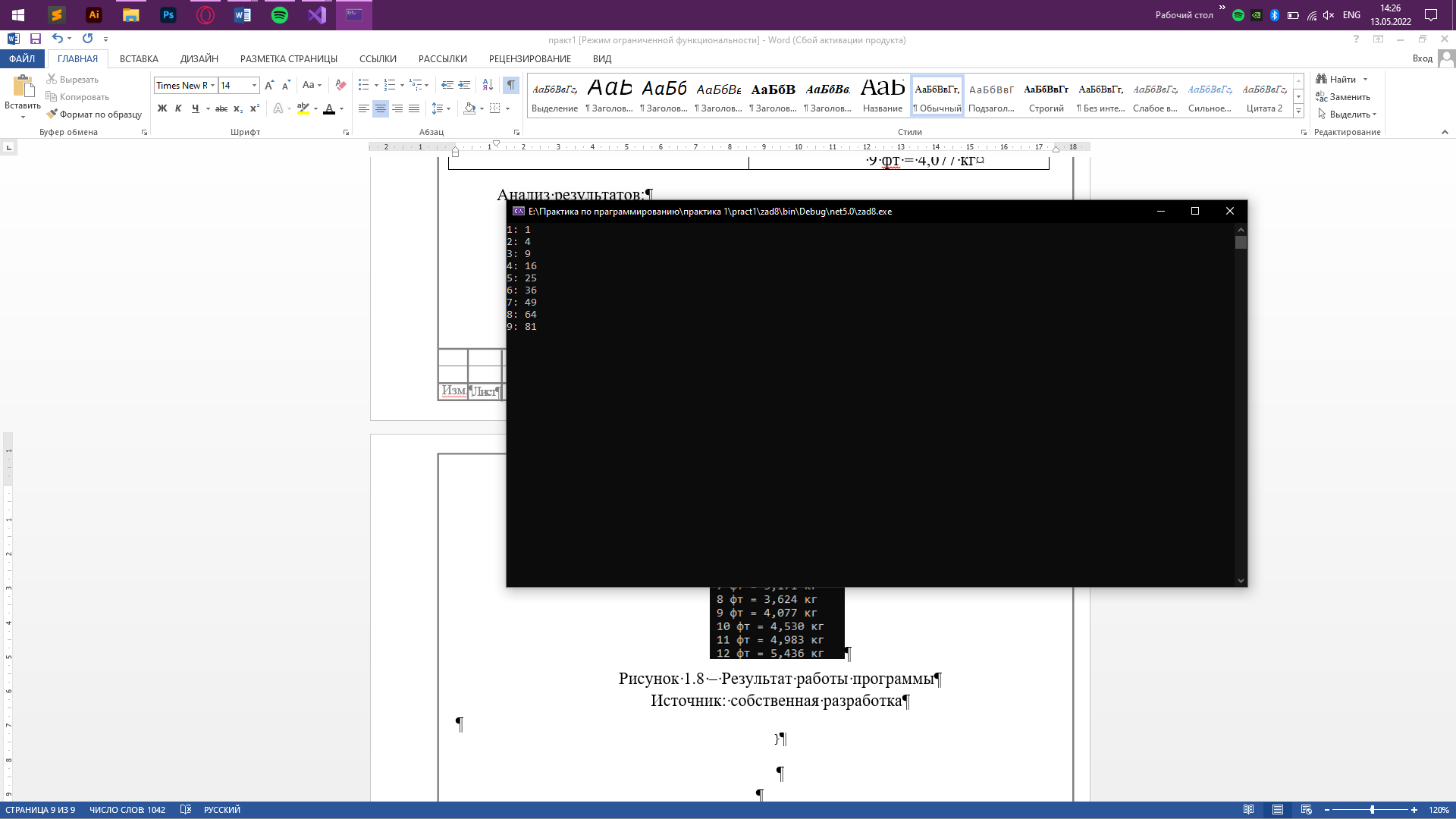


Рисунок 1.8 – Результат работы программы

Источник: собственная разработка

Задание 9. Табулирование функций. Составить программу вычисления значений функции F(*x*) на отрезке [A, B] в точках xi=x+H, где H=(B-A)/M, M – заданное целое число.

Функция: cos x

Параметры: A= ; B=.

M=20.

Листинг программы:

static double Func(double x)

{

return Math.Cos(x);

}

static void Main()

{

double a = Math.PI / 3;

double b = 2\*Math.PI/3;

double h = (b - a) / 20d;

for (double i = a; i <= b; i += h)

Console.WriteLine("Func({0:0.000}) = {1:0.000}", i, Func(i));

}

}

Таблица 1.9 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Функция: cos x  Параметры: A= ; B=.  M=20. | Func(1,047) = 0,500  Func(1,100) = 0,454  Func(1,152) = 0,407  Func(1,204) = 0,358  Func(1,257) = 0,309  Func(1,309) = 0,259 |

Анализ результатов:

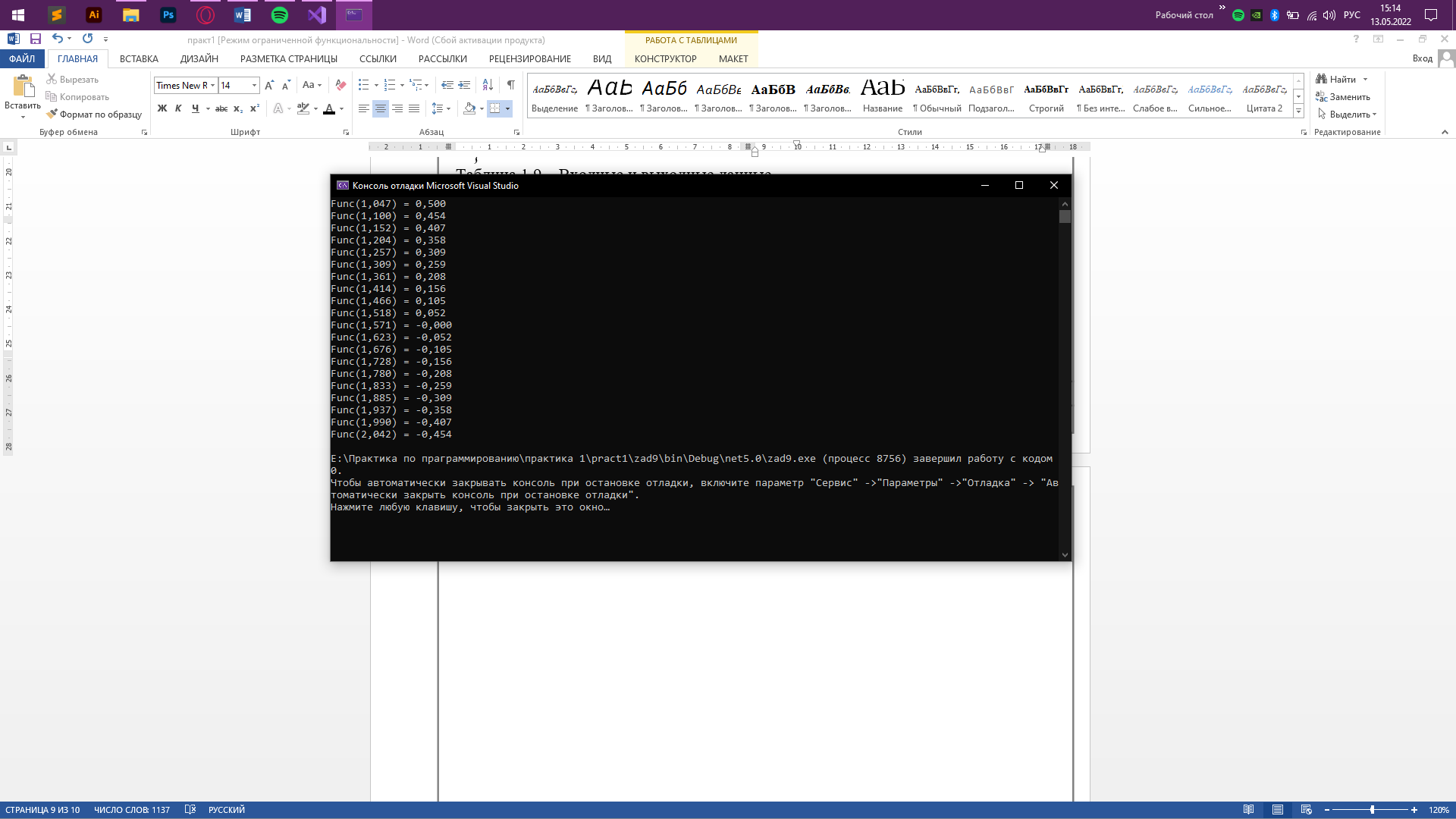


Рисунок 1.9 – Результат работы программы

Источник: собственная разработка

Задание 10.  Написать программу, находящая все двузначные числа, сумма квадратов которых кратна 13

Листинг программы:

static void Main(string[] args)

{

for (int i = 10; i < 100; i++)

{

if ((from x in i.ToString() select Math.Pow(Convert.ToInt32("" + x), 2)).Sum() % 13 == 0) Console.WriteLine(i);

}

Console.ReadKey();

}

}

Таблица 2.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Целые числа, сумма квадратов кратных 13 | 15  18  23  32  46  47 |

Анализ результатов:

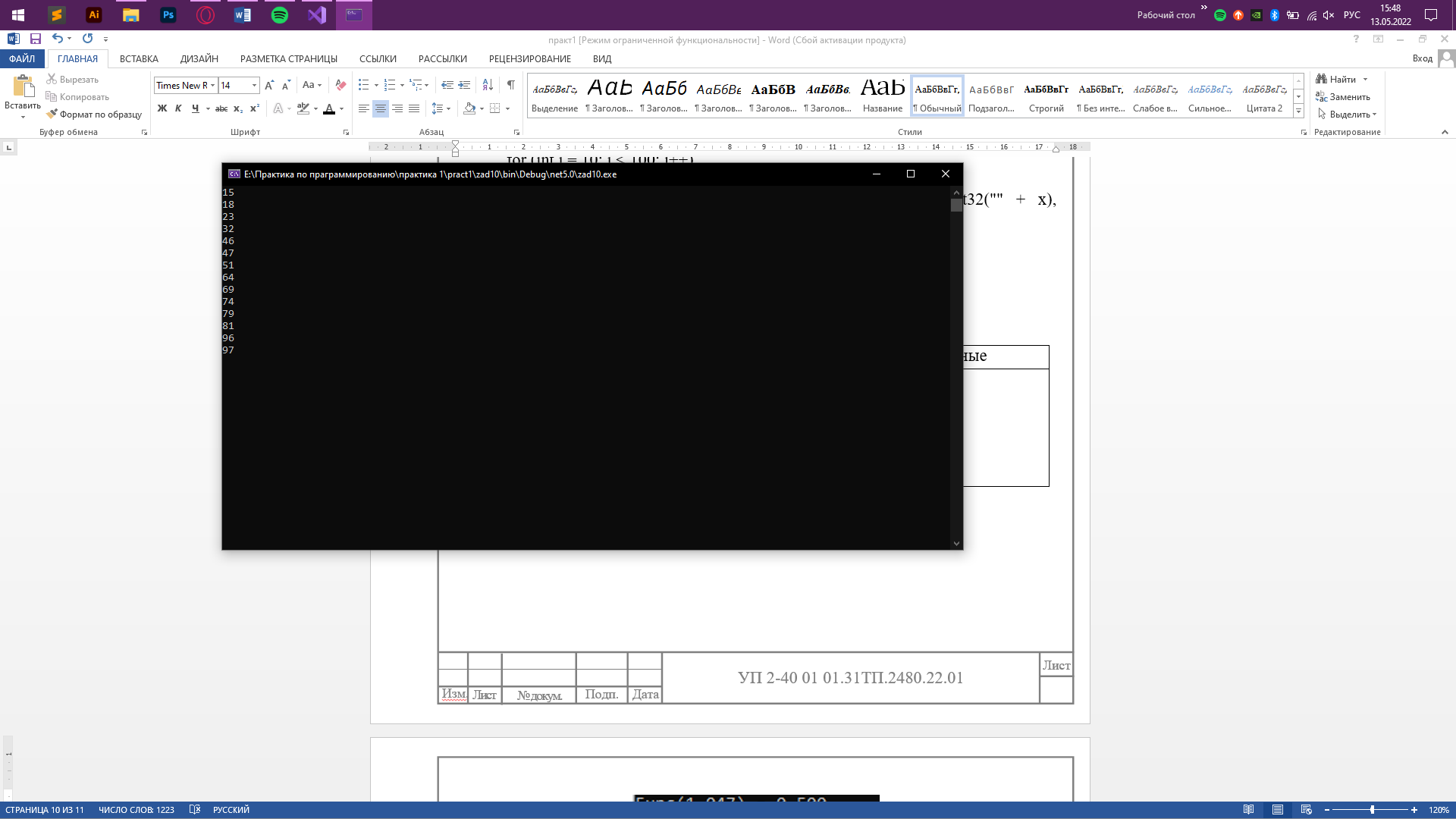


Рисунок 2.1 – Результат работы программы

Источник: собственная разработка