# **Лабораторная работа № 5-6**

**Тема:** Разработка и тестирование программного кода на языке С# с использованием

циклов с предусловием и постусловием.

**Цель работы:**

1. формирование навыков разработки и тестирования программ на языке С# с использованием циклов с предусловием и постусловием;
2. формирование навыков работы в среде Visual Studio.Net;
3. изучение операторов цикла с предусловием и постусловием для описания циклических алгоритмов на С#

. **Оборудование:**

1. Ноутбук MSI Modern
2. Программное обеспечение: ОС Windows, среда Visual Studio.Net

**Контрольные вопросы:**

1. Какие виды операторов цикла используются в языке C#? (Написать название и общий вид оператора)

В языке программирования C# используются следующие виды операторов цикла:

1. Цикл for. Общий вид оператора:

for (инициализация; условие; инкремент)

{

// Код для выполнения

}

2. Цикл while. Общий вид оператора:

while (условие)

{

// Код для выполнения

}

3. Цикл do-while. Общий вид оператора:

do

{

// Код для выполнения

}

while (условие);

4. Цикл foreach. Общий вид оператора:

foreach (тип переменная in коллекция)

{

// Код для выполнения

}

**2. Какие операторы C# используются для описания итерационных циклов?**

В C# для описания итерационных циклов используются следующие операторы:

1. Оператор for: Используется для управления циклом с определенным количеством итераций. Он состоит из инициализации, условия продолжения и шага итерации.

2. Оператор while: Используется для управления циклом, который выполняется, пока условие истинно. Условие проверяется перед каждой итерацией.

3. Оператор do-while: Используется для управления циклом, который выполняется, пока условие истинно. Условие проверяется после каждой итерации.

4. Оператор foreach: Используется для итерации по элементам коллекции, такой как массив или список. Он автоматически перебирает все элементы коллекции.

5. Оператор break: Используется для прерывания выполнения цикла и перехода к следующей выполняемой строчке кода за пределами цикла.

6. Оператор continue: Используется для пропуска текущей итерации цикла и перехода к следующей итерации.

**3. Как выполняется оператор цикла с постусловием в языке C#?**

В языке C# оператор цикла с постусловием представляется с помощью оператора do-while. Он имеет следующий синтаксис:

do

{

// выполняемый код

}

while (условие);

Оператор do-while выполняет блок кода сначала, а затем проверяет условие. Если условие истинно, то цикл повторяется и блок кода выполняется снова. Процесс повторяется до тех пор, пока условие не станет ложным. Таким образом, в отличие от оператора while, который проверяет условие перед каждой итерацией, оператор do-while гарантирует выполнение блока кода хотя бы один раз.

**4. Когда при использовании оператора цикла с постусловием цикл будет бесконечным?**

Цикл с постусловием (do-while) будет бесконечным, если условие, проверяемое после каждой итерации цикла, всегда остаётся истинным. В этом случае, блок кода внутри цикла будет выполняться повторно без возможности завершения.

Вот пример бесконечного цикла с постусловием:

do

{

// выполняемый код

}

while (true);

В данном примере условие true всегда будет истинным, поэтому цикл будет выполняться бесконечно. Чтобы избежать бесконечного цикла, необходимо обеспечить условие, которое станет ложным в какой-то момент, позволяя выйти из цикла.

**5. Cколько раз выполнится цикл с постусловием, если условие ложно?**

Цикл с постусловием, такой как do-while в C#, выполнится минимум один раз, даже если условие ложно. Это связано с тем, что проверка условия осуществляется после выполнения тела цикла.

**6. Чем отличается оператор цикла с предусловием от оператора цикла с постусловием.**

Оператор цикла с предусловием, такой как while в C#, проверяет условие перед выполнением тела цикла. Если условие ложно с самого начала, тело цикла не исполнится ни разу.

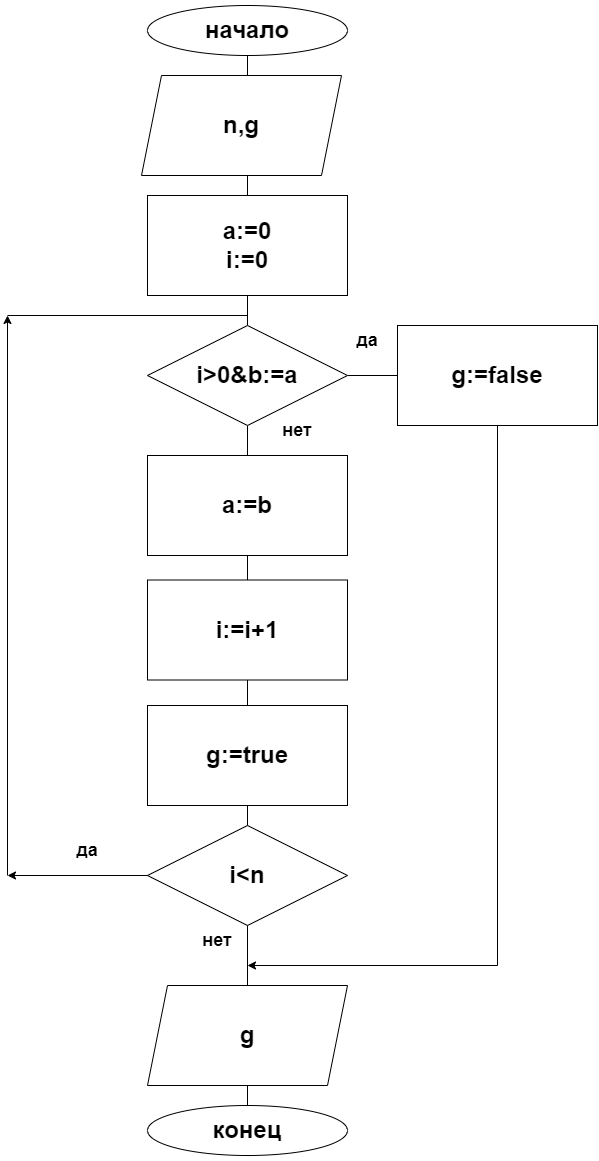
Оператор цикла с постусловием, то есть do-while, сначала выполняет тело цикла, а затем проверяет условие. Это означает, что тело цикла выполнится минимум один раз, даже если условие ложно с самого начала.

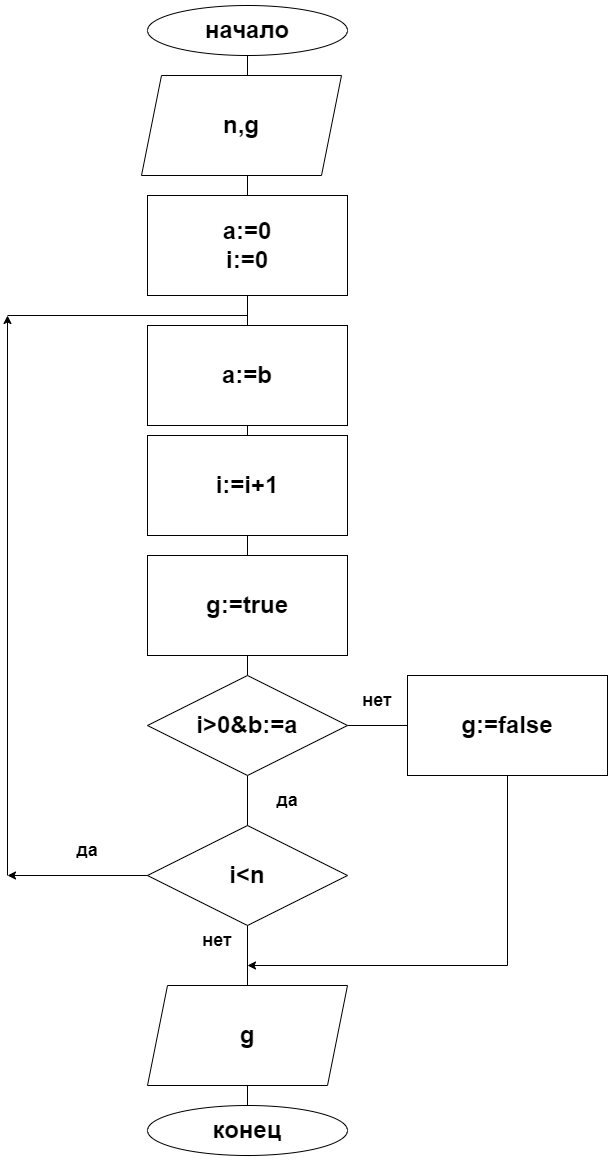
**Ход работы:**

1. Разработать алгоритм задачи и представить его в виде схемы программы
2. Используя Visual Studio.Net создать файл с программой
3. Выполнить тестирование и отладку программу
4. Результаты представить в виде отчета
5. Сделать вывод о проделанной работе

**Задание на лабораторную работу:** Дано n вещественных чисел. Определить, образуют ли они знакочередующуюся последовательность.

**Схема программы**



****

**Листинг с исходным кодом**

*Код с циклом for*

using System;

class Program

{

static void Main(string[] args)

{

Console.Write("Введите количество чисел: ");

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

double a = 0;

double b = 0;

bool g = true;

for (int i = 0; i < n; i++)

{

Console.Write($"Введите число {i + 1}: ");

b = Convert.ToDouble(Console.ReadLine());

if (i > 0 && Math.Sign(b) == Math.Sign(a))

{

g = false;

break;

}

a = b;

}

if (g)

Console.WriteLine("Числа образуют знакочередующуюся последовательность.");

else

Console.WriteLine("Числа не образуют знакочередующуюся последовательность.");

Console.ReadKey();

}

}

*Код с циклом while*

using System;

class Program

{

static void Main(string[] args)

{

Console.Write("Введите количество чисел: ");

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

double a = 0;

double b = 0;

bool g = true;

int i = 0;

while (i < n)

{

Console.Write($"Введите число {i + 1}: ");

b = Convert.ToDouble(Console.ReadLine());

if (i > 0 && Math.Sign(b) == Math.Sign(a))

{

g = false;

break;

}

a = b;

i++;

}

if (g)

Console.WriteLine("Числа образуют знакочередующуюся последовательность.");

else

Console.WriteLine("Числа не образуют знакочередующуюся последовательность.");

Console.ReadKey();

}

}

*Код с циклом do….while*

using System;

class Program

{

static void Main(string[] args)

{

Console.Write("Введите количество чисел: ");

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

double a = 0;

double b = 0;

bool g = true;

int i = 0;

do

{

Console.Write($"Введите число {i + 1}: ");

b = Convert.ToDouble(Console.ReadLine());

if (i > 0 && Math.Sign(b) == Math.Sign(a))

{

g = false;

break;

}

a = b;

i++;

} while (i < n);

if (g)

Console.WriteLine("Числа образуют знакочередующуюся последовательность.");

else

Console.WriteLine("Числа не образуют знакочередующуюся последовательность.");

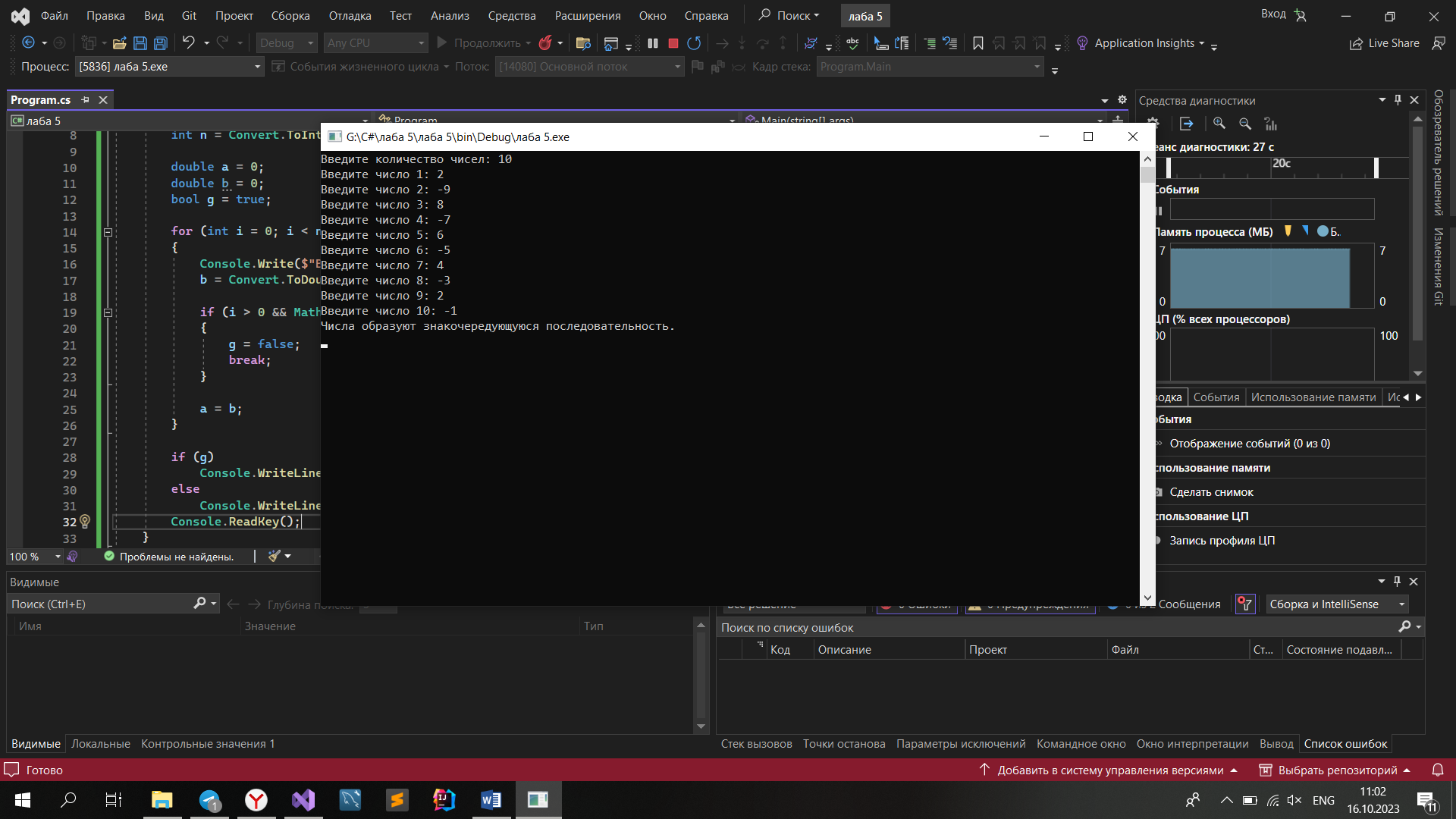
Console.ReadKey();

}

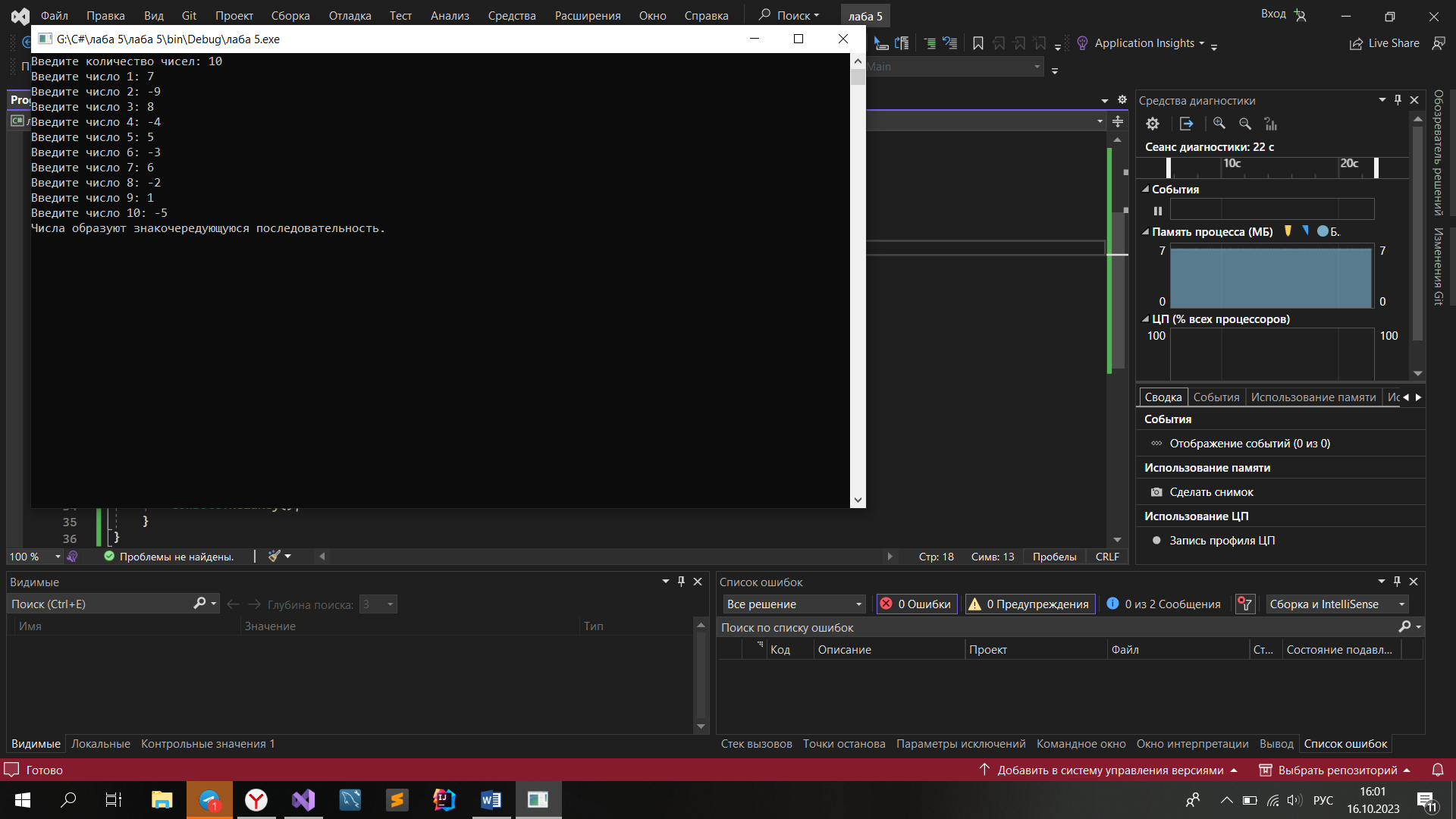
}

**Результаты тестирования**

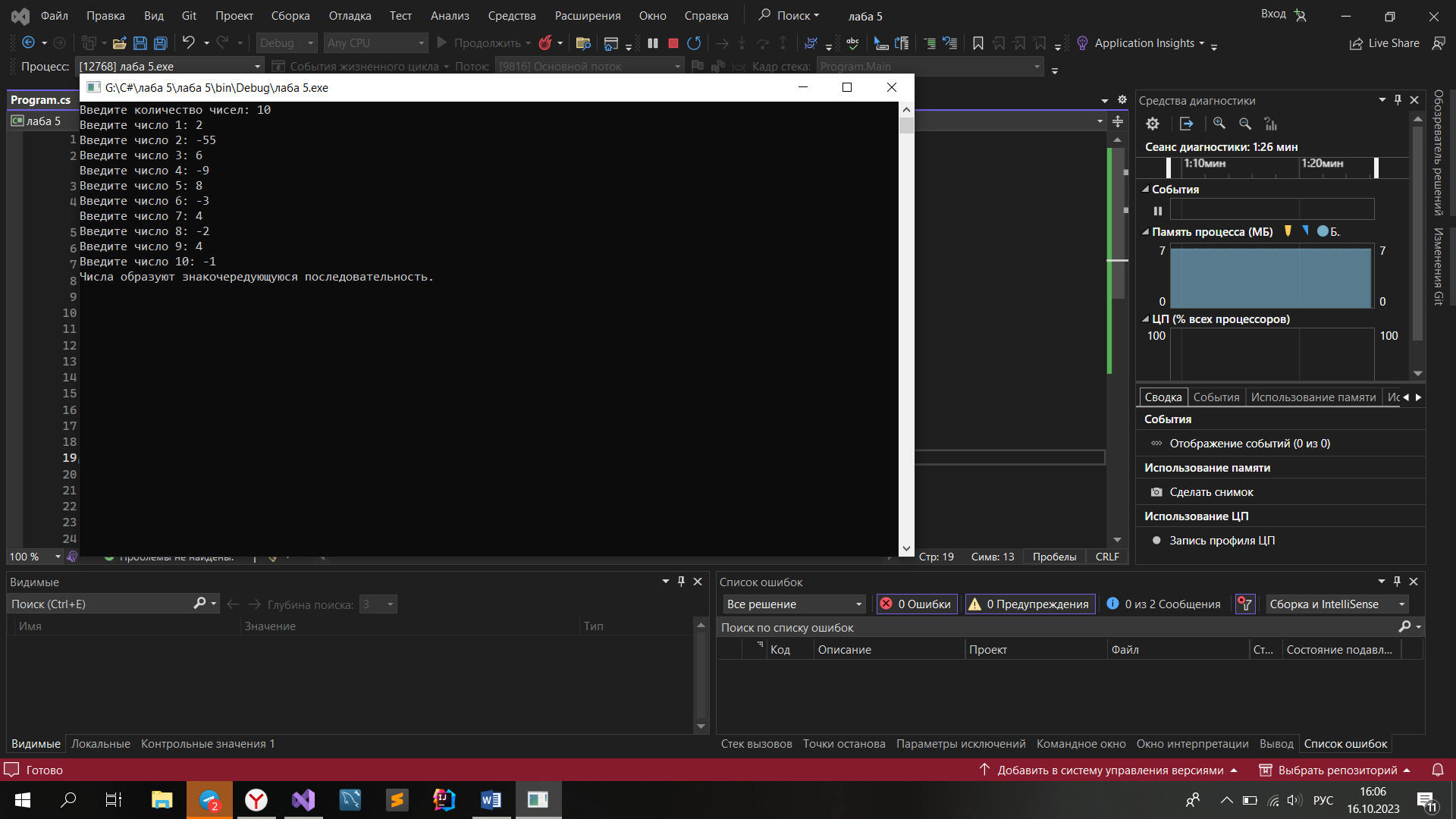
*Код с циклом for*



*Код с циклом while*



*Код с циклом do….while*



**Вывод по работе.**

В ходе работы были изучены и применены алгоритмы с циклом в контексте программирования на языке C#. Эти алгоритмы, использующие условные операторы и циклы, являются фундаментальными элементами большинства программ. Разработка и тестирование кода на языке C# с использованием алгоритмов с циклом помогает создавать эффективные и надежные приложения.