**Пензенский государственный университет**

**Кафедра вычислительной техники**

**Отчет**

по лабораторной работе № 1

по дисциплине: “Логика и алгоритмизация в инженерных задачах"

Выполнили студенты группы 22ВВП2:

Бормотов А.А.

Кузьмин Д.В.

Кочетков А.М.

Приняли:

Юрова О.В.

Акифьев И.В.

2023

**Цель**: Выполнить задания на работу с простыми структурами данных.

**Задача 1**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace wow

{

internal class Program

{

static void Main(string[] args)

{

int[] Array = {10, 0, -10, 33, - 54};

int Min\_Element = Array[0];

int Max\_Element = Array[0];

for (int Index = 0; Index < Array.Length; Index++)

{

if (Array[Index] > Max\_Element)

{

Max\_Element = Array[Index];

}

if (Array[Index] < Min\_Element)

{

Min\_Element = Array[Index];

}

}

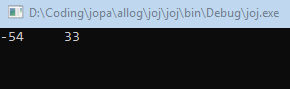
Console.WriteLine(Min\_Element + "\t" + Max\_Element);

}

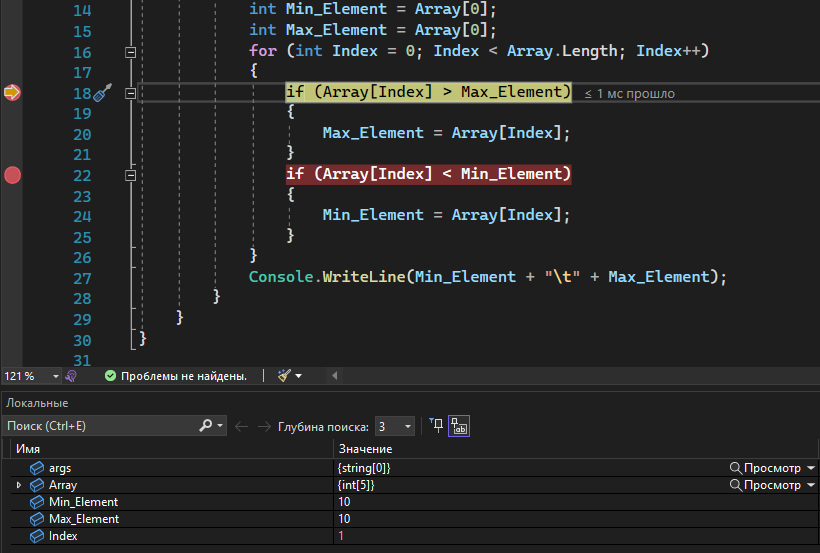
}

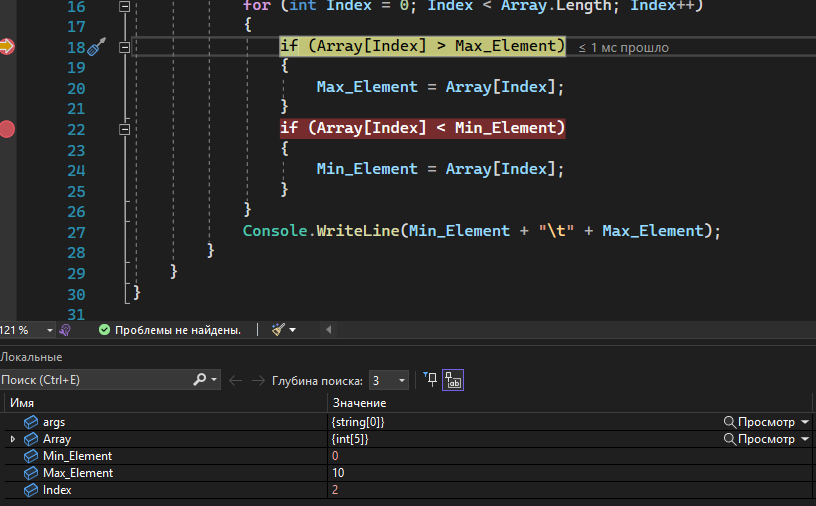
}

**Результат работы программы:**



**Трассировка программы:**





**Задача 2**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ArrayInit

{

internal class Program

{

static void Main(string[] args)

{

int[] Array = new int[20];

Random Random\_Num = new Random();

for (int Index = 0; Index < Array.Length; Index++)

{

int Number = Random\_Num.Next(0, 100);

Array[Index] = Number;

Console.Write(Array[Index] + " ");

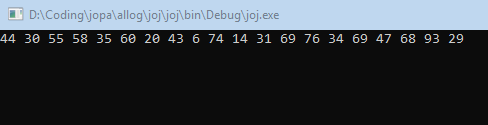
}

}

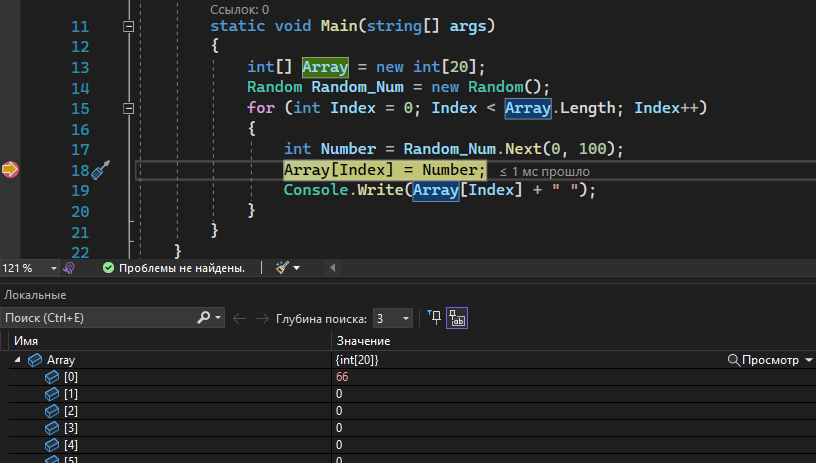
}

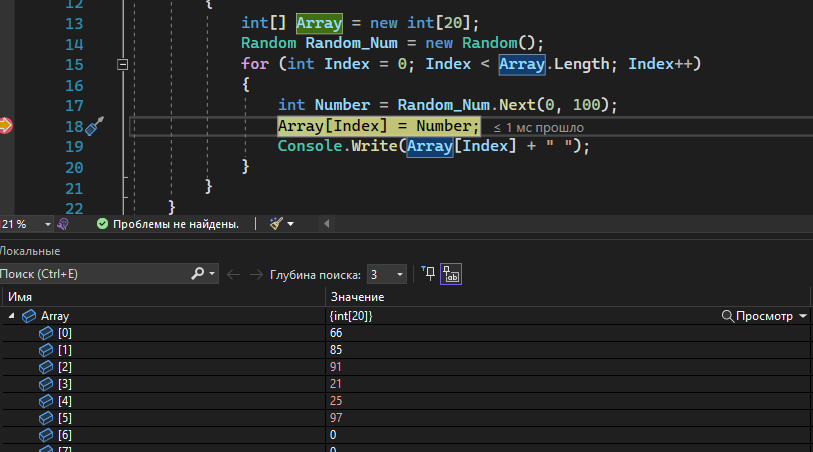
}

**Результат работы программы:**



**Трассировка программы:**





**Задача 3**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ArrayFromKB

{

internal class Program

{

static void Main(string[] args)

{

int[] Array = new int[int.Parse(Console.ReadLine())];

Random Random\_Num = new Random();

for (int Index = 0; Index < Array.Length; Index++)

{

int Number = Random\_Num.Next(0, 100);

Array[Index] = Number;

Console.Write(Array[Index] + " ");

}

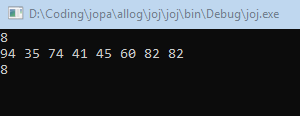
Console.WriteLine("\n" + Array.Length);

}

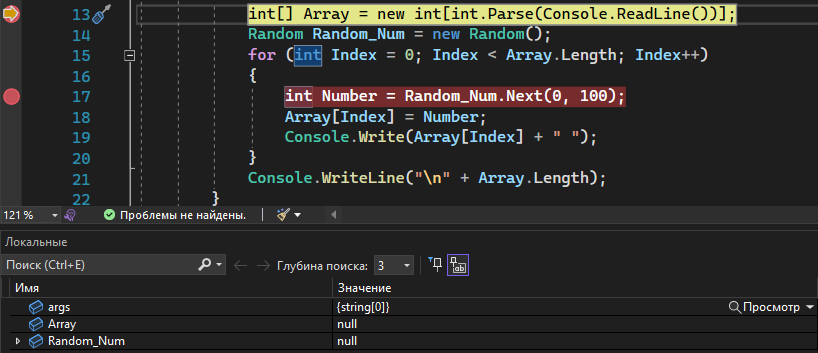
}

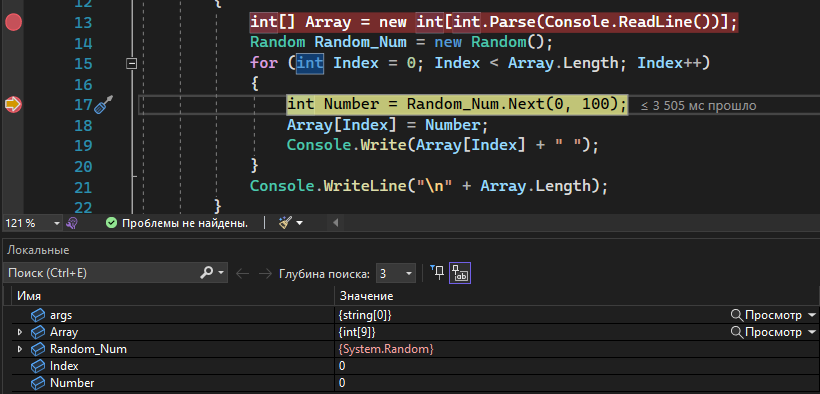
}

**Результат работы программы:**



**Трассировка программы:**





**Задача 4**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace SummColonArray

{

internal class Program

{

static void Main(string[] args)

{

int[,] Array = new int[5,7];

int[] Summ\_Array\_String = new int[5];

int[] Summ\_Array\_Colon = new int[7];

Random Random\_Num = new Random();

int Summ = 0;

for (int F\_Index = 0; F\_Index < 5; F\_Index++) // Fill array

{

for (int S\_Index = 0; S\_Index < 7; S\_Index++)

{

int Number = Random\_Num.Next(0, 100);

Array[F\_Index, S\_Index] = Number;

Console.Write(Array[F\_Index, S\_Index] + " ");

}

Console.WriteLine();

}

for (int F\_Index = 0; F\_Index < 5; F\_Index++) // Summ of each string of array

{

for (int S\_Index = 0; S\_Index < 7; S\_Index++)

{

Summ += Array[F\_Index, S\_Index];

}

Summ\_Array\_String[F\_Index] = Summ;

}

for (int S\_Index = 0; S\_Index < 7; S\_Index++) // Summ of each colon of array

{

Summ = 0;

for (int F\_Index = 0; F\_Index < 5; F\_Index++)

{

Summ += Array[F\_Index, S\_Index];

}

Summ\_Array\_Colon[S\_Index] = Summ;

}

Console.WriteLine();

for (int Index = 0; Index < Summ\_Array\_String.Length; Index++) // Displaying arrays for debug

{

Console.WriteLine(Summ\_Array\_String[Index]);

}

Console.WriteLine();

for (int Index = 0; Index < Summ\_Array\_Colon.Length; Index++)

{

Console.WriteLine(Summ\_Array\_Colon[Index]);

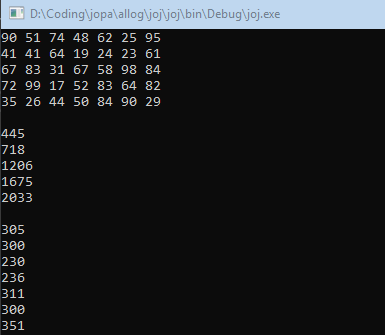
}

}

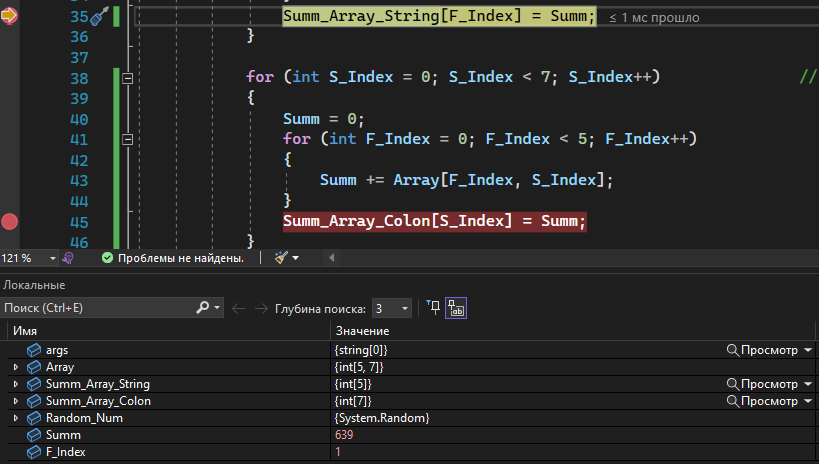
}

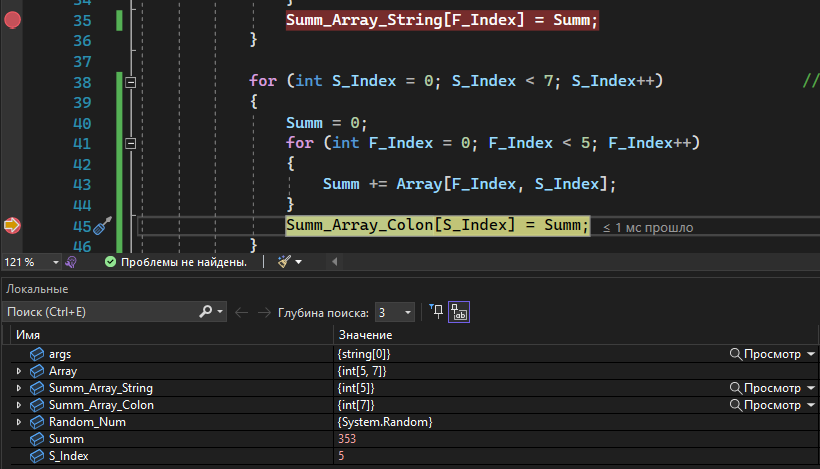
}

**Результат работы программы:**



**Трассировка программы:**





**Задача 5**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Linq;

using System.Reflection;

using System.Security.Cryptography.X509Certificates;

using System.Text;

using System.Threading.Tasks;

namespace StructSearch

{

public struct Student

{

public string first\_name;

public string last\_name;

public string age;

}

internal class Program

{

static void Main(string[] args)

{

List<Student> StructArray = new List<Student>();

int Index\_Count = 0;

Student S1 = new Student //Creating structures and saving them into array

{

first\_name = "Toni",

last\_name = "Bor",

age = "19"

};

StructArray.Add(S1);

Student S2 = new Student

{

first\_name = "Astrex",

last\_name = "Exl",

age = "16"

};

StructArray.Add(S2);

Student S3 = new Student

{

first\_name = "Dani",

last\_name = "Lakom",

age = "19"

};

StructArray.Add(S3);

Student S4 = new Student

{

first\_name = "Dani",

last\_name = "Kuzya",

age = "19"

};

StructArray.Add(S4);

Student S5 = new Student

{

first\_name = "Dani",

last\_name = "Lakom",

age = "21"

};

StructArray.Add(S5);

Console.WriteLine("Input first name of student(press enter to skip): "); //Filter by first name

string Param\_FName = Console.ReadLine();

for (int Index = 0; Index < StructArray.Count; Index++)

{

if (StructArray[Index].first\_name == Param\_FName)

{

StructArray[Index\_Count] = StructArray[Index];

Index\_Count++;

}

}

if (Param\_FName != "")

{

for (int Index = StructArray.Count - 1; Index >= Index\_Count; Index--)

{

StructArray.RemoveAt(Index);

}

Index\_Count = 0;

}

Console.WriteLine("Input last name of student(press enter to skip): "); //Filter by last name

string Param\_LName = Console.ReadLine();

for (int Index = 0; Index < StructArray.Count; Index++)

{

if (StructArray[Index].last\_name == Param\_LName)

{

StructArray[Index\_Count] = StructArray[Index];

Index\_Count++;

}

}

if (Param\_LName != "")

{

for (int Index = StructArray.Count - 1; Index >= Index\_Count; Index--)

{

StructArray.RemoveAt(Index);

}

Index\_Count = 0;

}

Console.WriteLine("Input age of student(press enter to skip): "); //Filter by age

string Param\_Age = Console.ReadLine();

for (int Index = 0; Index < StructArray.Count; Index++)

{

if (StructArray[Index].age == Param\_Age)

{

StructArray[Index\_Count] = StructArray[Index];

Index\_Count++;

}

}

if (Param\_Age != "")

{

for (int Index = StructArray.Count - 1; Index >= Index\_Count; Index--)

{

StructArray.RemoveAt(Index);

}

}

if (StructArray.Count != 0 && (Param\_FName != "" || Param\_LName != "" || Param\_Age != "")) //Console output for debugging purposes

{

Console.WriteLine("Success! This what the algorithm had found: ");

for (int Index = 0; Index < StructArray.Count; Index++)

{

Console.WriteLine(StructArray[Index].first\_name + " " + StructArray[Index].last\_name + " " + StructArray[Index].age);

}

}

if (StructArray.Count == 0 || Param\_FName == "" && Param\_LName == "" && Param\_Age == "")

{

Console.WriteLine("Error! Algorithm couldn't find anything with such parameters");

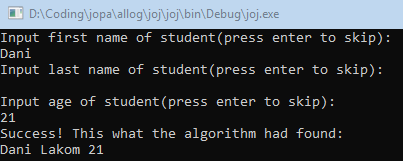
}

}

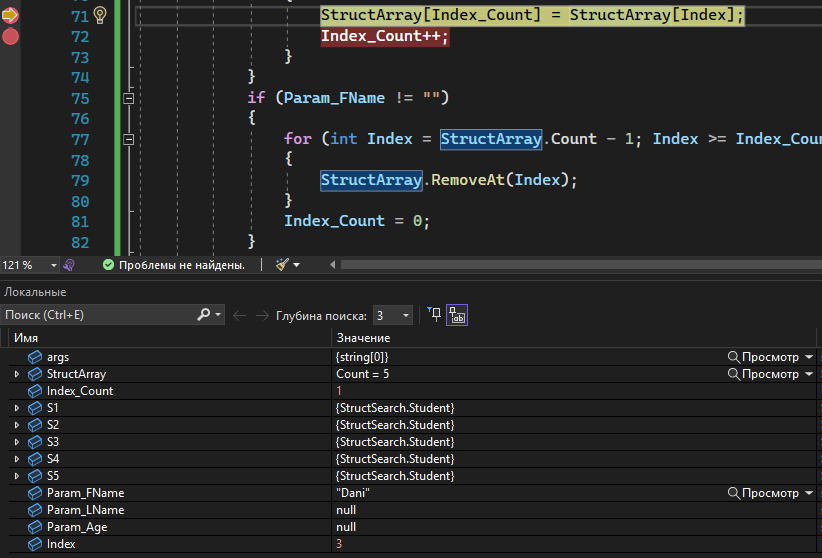
}

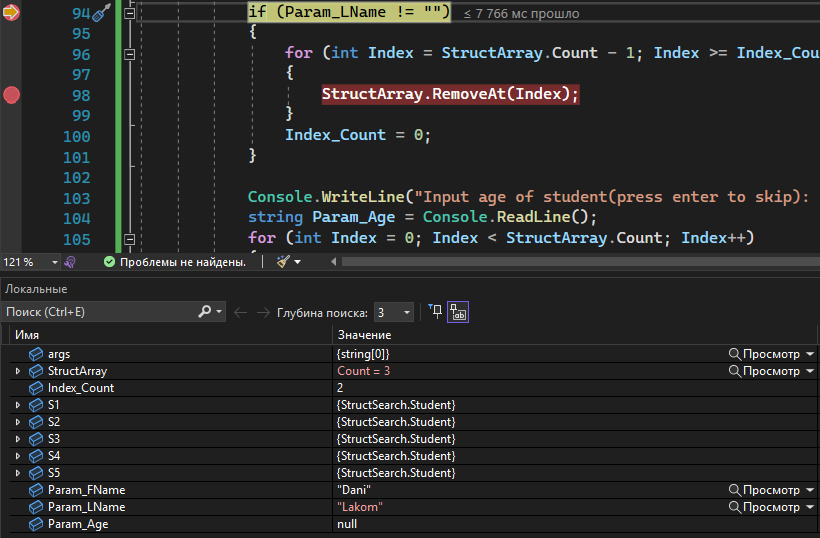
}

**Результат работы программы:**



**Трассировка программы:**





**Вывод**: Выполнили задания на работу с простыми структурами данных, их сортировку, заполнения случайными числами, поиск элементов по параметрам и выполнение действиями над ними.