МИНИСТЕРСТВОНАУКИИВЫСШЕГООБРАЗОВАНИЯ РОССИЙСКОЙФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский государственный университет им. А.Н. Косыгина (Технологии. Дизайн. Искусство)»

Кафедра автоматизированных систем обработки информации и управления

Отчет по лабораторной работе №1

по дисциплине «Информационные системы и базы данных».

Тема: «Обработка одномерных массивов»

Выполнил: Юрков Д.А., группа МВА-122 Проверил: Адаев Р.Б.

Москва 2025

Задание 22:

Дан одномерный массив Xn. Найти количество элементов массива, значение которых больше величины А. Сформировать новый массив из элементов массива Xn, принадлежащих отрезку (В, С).

Контрольный пример: одномерный массив c длиной 6: {1,2,3,4,5,6}

Количество элементов, больших A: 3

Элементы, принадлежащие отрезку (B, C): {4, 5}

Блок-схема алгоритма представлена на рисунке 1.

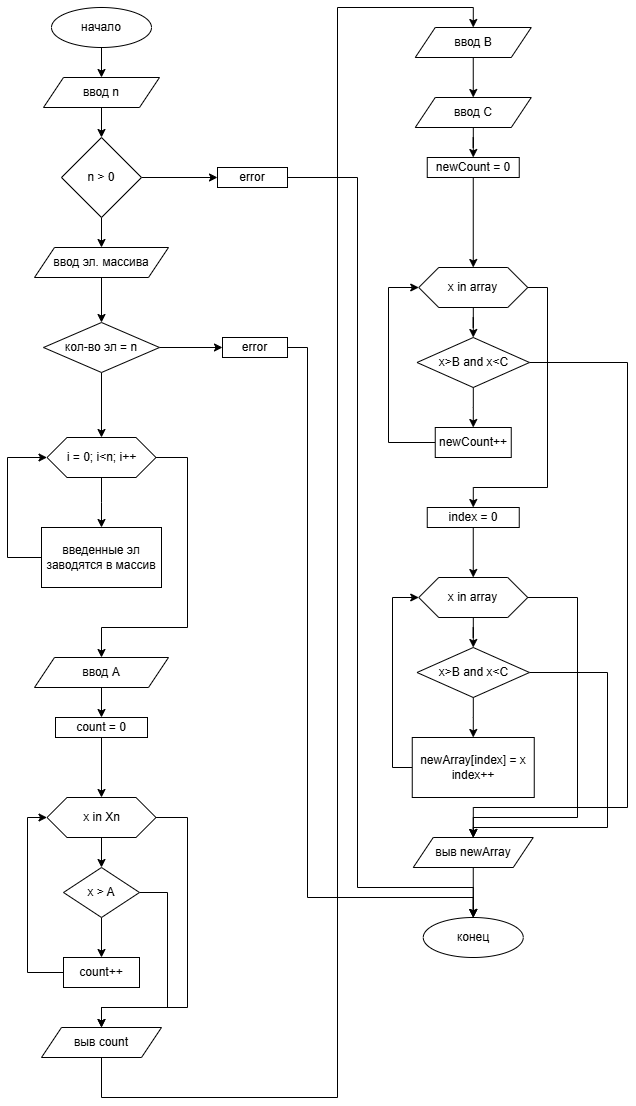


Рис.1 Блок-схема алгоритма

Код консольного приложения:

using System;

class Program

{

static void Main()

{

try

{

Console.WriteLine("Введите длину массива:");

int n = int.Parse(Console.ReadLine());

if (n <= 0)

{

Console.WriteLine("Ошибка: длина массива должна быть положительным числом.");

return;

}

Console.WriteLine($"Введите {n} элементов массива, разделенных пробелами:");

string input = Console.ReadLine();

string[] parts = input.Split(' ');

if (parts.Length != n)

{

Console.WriteLine($"Ошибка: введено {parts.Length} элементов, ожидалось {n}.");

return;

}

double[] Xn = new double[n];

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

{

Xn[i] = double.Parse(parts[i]);

}

Console.WriteLine("Введите значение A:");

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

int count = 0;

foreach (double x in Xn)

{

if (x > A) count++;

}

Console.WriteLine("Количество элементов, больших A: " + count);

Console.WriteLine("Введите значение B:");

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

Console.WriteLine("Введите значение C:");

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

int newCount = 0;

foreach (double x in Xn)

{

if (x > B && x < C) newCount++;

}

double[] newArray = new double[newCount];

int index = 0;

foreach (double x in Xn)

{

if (x > B && x < C)

{

newArray[index] = x;

index++;

}

}

Console.WriteLine("Новый массив:");

Console.WriteLine(string.Join(" ", newArray));

}

catch (FormatException)

{

Console.WriteLine("Ошибка: введены некорректные данные. Пожалуйста, используйте числа.");

}

catch (Exception ex)

{

Console.WriteLine($"Произошла ошибка: {ex.Message}");

}

}

}

Результат выполнения консольного приложения:

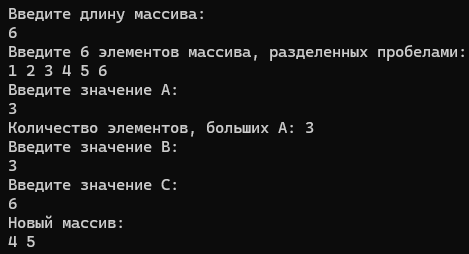


Рис. 2. Результат выполнения консольной программы

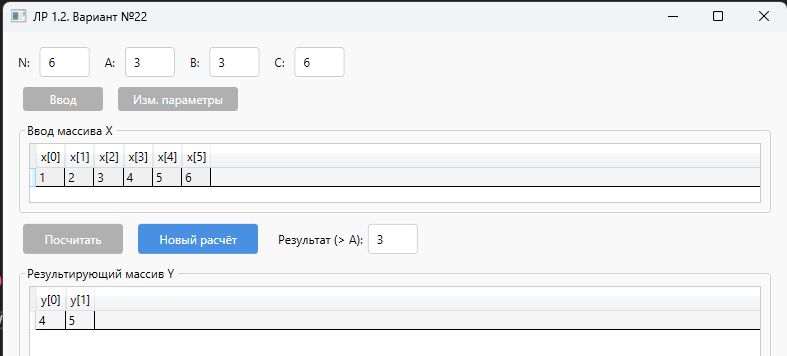


Рис.3 Результат выполнения Windows Form

Код WPF:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Windows;

using System.Windows.Controls;

namespace ISDBlab2

{

public partial class MainWindow : Window

{

private int N;

private double A, B, C;

private int[] x;

public MainWindow()

{

InitializeComponent();

InitializeForm();

}

private void InitializeForm()

{

PanelInput.Visibility = Visibility.Collapsed;

PanelResult.Visibility = Visibility.Collapsed;

BtnCalculate.IsEnabled = false;

}

private void BtnInput\_Click(object sender, RoutedEventArgs e)

{

if (!int.TryParse(TxtN.Text, out N) || N <= 0)

{

MessageBox.Show("Введите корректное положительное N", "Ошибка");

return;

}

if (!double.TryParse(TxtA.Text, out A) ||

!double.TryParse(TxtB.Text, out B) ||

!double.TryParse(TxtC.Text, out C))

{

MessageBox.Show("Введите корректные числовые параметры A, B, C", "Ошибка");

return;

}

x = new int[N];

DataGridInput.Columns.Clear();

var row = new Dictionary<string, int>();

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

{

DataGridInput.Columns.Add(new DataGridTextColumn

{

Header = $"x[{i}]",

Binding = new System.Windows.Data.Binding($"[{i}]") { Mode = System.Windows.Data.BindingMode.TwoWay }

});

row[i.ToString()] = 0;

}

DataGridInput.ItemsSource = new List<Dictionary<string, int>> { row };

PanelInput.Visibility = Visibility.Visible;

BtnCalculate.IsEnabled = true;

PanelResult.Visibility = Visibility.Collapsed;

}

private void BtnCalculate\_Click(object sender, RoutedEventArgs e)

{

var row = (Dictionary<string, int>)DataGridInput.Items[0];

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

{

if (!row.TryGetValue(i.ToString(), out int val))

{

MessageBox.Show($"Значение x[{i}] не задано", "Ошибка");

return;

}

x[i] = val;

}

int countGtA = x.Count(v => v > A);

TxtResult.Text = countGtA.ToString();

var y = x.Where(v => v > B && v < C).ToArray();

if (y.Length == 0)

{

MessageBox.Show("Массив Y не сформирован: нет элементов в (B, C)", "Внимание");

return;

}

DataGridOutput.Columns.Clear();

var outRow = new Dictionary<string, int>();

for (int i = 0; i < y.Length; i++)

{

DataGridOutput.Columns.Add(new DataGridTextColumn

{

Header = $"y[{i}]",

Binding = new System.Windows.Data.Binding($"[{i}]")

});

outRow[i.ToString()] = y[i];

}

DataGridOutput.ItemsSource = new List<Dictionary<string, int>> { outRow };

PanelResult.Visibility = Visibility.Visible;

BtnInput.IsEnabled = false;

BtnReset.IsEnabled = false;

BtnCalculate.IsEnabled = false;

}

private void BtnNew\_Click(object sender, RoutedEventArgs e)

{

TxtN.Text = TxtA.Text = TxtB.Text = TxtC.Text = TxtResult.Text = "";

DataGridInput.Columns.Clear();

DataGridInput.ItemsSource = null;

DataGridOutput.Columns.Clear();

DataGridOutput.ItemsSource = null;

BtnInput.IsEnabled = true;

BtnReset.IsEnabled = true;

InitializeForm();

}

}

}

XAML:

<Window x:Class="ISDBlab2.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Title="ЛР 1.2. Вариант №22" Height="600" Width="800"

Background="#F9F9F9">

<Window.Resources>

<Style TargetType="Button">

<Setter Property="Background" Value="#4A90E2"/>

<Setter Property="Foreground" Value="White"/>

<Setter Property="Padding" Value="8,4"/>

<Setter Property="Margin" Value="5,0"/>

<Setter Property="BorderThickness" Value="0"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="Button">

<Border Background="{TemplateBinding Background}"

CornerRadius="3"

Padding="{TemplateBinding Padding}">

<ContentPresenter HorizontalAlignment="Center" VerticalAlignment="Center"/>

</Border>

<ControlTemplate.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="Background" Value="#357ABD"/>

</Trigger>

<Trigger Property="IsEnabled" Value="False">

<Setter Property="Background" Value="#B0B0B0"/>

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<Style TargetType="TextBox">

<Setter Property="Background" Value="White"/>

<Setter Property="BorderBrush" Value="#CCCCCC"/>

<Setter Property="BorderThickness" Value="1"/>

<Setter Property="Padding" Value="4"/>

<Setter Property="VerticalContentAlignment" Value="Center"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="TextBox">

<Border Background="{TemplateBinding Background}"

BorderBrush="{TemplateBinding BorderBrush}"

BorderThickness="{TemplateBinding BorderThickness}"

CornerRadius="3">

<ScrollViewer x:Name="PART\_ContentHost" Margin="2"/>

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<Style TargetType="GroupBox">

<Setter Property="BorderBrush" Value="#CCCCCC"/>

<Setter Property="Padding" Value="5"/>

</Style>

<Style TargetType="DataGrid">

<Setter Property="Background" Value="White"/>

<Setter Property="BorderBrush" Value="#CCCCCC"/>

<Setter Property="BorderThickness" Value="1"/>

<Setter Property="RowBackground" Value="#F5F5F5"/>

<Setter Property="AlternatingRowBackground" Value="White"/>

</Style>

</Window.Resources>

<Grid Margin="15">

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<StackPanel Orientation="Horizontal" Grid.Row="0" Margin="0,0,0,10">

<TextBlock Text="N:" VerticalAlignment="Center" Margin="0,0,5,0"/>

<TextBox x:Name="TxtN" Width="50" Margin="5,0,15,0"/>

<TextBlock Text="A:" VerticalAlignment="Center" Margin="0,0,5,0"/>

<TextBox x:Name="TxtA" Width="50" Margin="5,0,15,0"/>

<TextBlock Text="B:" VerticalAlignment="Center" Margin="0,0,5,0"/>

<TextBox x:Name="TxtB" Width="50" Margin="5,0,15,0"/>

<TextBlock Text="C:" VerticalAlignment="Center" Margin="0,0,5,0"/>

<TextBox x:Name="TxtC" Width="50" Margin="5,0,0,0"/>

</StackPanel>

<StackPanel Orientation="Horizontal" Grid.Row="1" Margin="0,0,0,10">

<Button x:Name="BtnInput" Content="Ввод" Width="80" Click="BtnInput\_Click"/>

<Button x:Name="BtnReset" Content="Изм. параметры" Width="120" Margin="10,0,0,0" Click="BtnNew\_Click"/>

</StackPanel>

<GroupBox Header="Ввод массива X" Grid.Row="2" Margin="0,0,0,10"

x:Name="PanelInput" Visibility="Collapsed">

<DataGrid x:Name="DataGridInput" AutoGenerateColumns="False" CanUserAddRows="False" Height="60"/>

</GroupBox>

<StackPanel Orientation="Horizontal" Grid.Row="3" Margin="0,0,0,10">

<Button x:Name="BtnCalculate" Content="Посчитать" Width="100" Click="BtnCalculate\_Click"/>

<Button x:Name="BtnNew" Content="Новый расчёт" Width="120" Margin="10,0,0,0" Click="BtnNew\_Click"/>

<TextBlock Text="Результат (> A):" VerticalAlignment="Center" Margin="20,0,5,0"/>

<TextBox x:Name="TxtResult" Width="50" IsReadOnly="True"/>

</StackPanel>

<GroupBox Header="Результирующий массив Y" Grid.Row="4" x:Name="PanelResult" Visibility="Collapsed">

<DataGrid x:Name="DataGridOutput" AutoGenerateColumns="False" CanUserAddRows="False"/>

</GroupBox>

</Grid>

</Window>