Министерство науки и высшего образования Российской Федерации

Федеральное государственное образовательное учреждение

высшего образования

**ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

Институт математики и информационных систем

Факультет автоматики и вычислительной техники

Кафедра систем автоматизации управления

**Дисциплина: Алгоритмы и структуры данных**

**Отчёт по лабораторной работе №4**

**Тема: «Одномерные и двумерные массивы»**

**Вариант 2, 7**

|  |  |  |  |
| --- | --- | --- | --- |
| Выполнил студент  группы ИТб 2302-02-20 |  | Ердяков Роман Александрович |  |
|  |  |
| Проверила |  | Кашина Елена Вячеславовна |  |

СОДЕРЖАНИЕ

[1 Задание 3](#_Toc200808731)

[2 Решение (задача 1) 4](#_Toc200808732)

[3 Тестирование (задача 1) 5](#_Toc200808733)

[4 Программа на C# (задача 1): 6](#_Toc200808734)

[5 Решение (задача 2) 7](#_Toc200808735)

[6 Тестирование (задача 2) 8](#_Toc200808736)

[7 Программа на C# (задача 2): 9](#_Toc200808737)

[8 Решение (задача 3) 10](#_Toc200808738)

[9 Тестирование (задача 3) 11](#_Toc200808739)

[10 Программа на C# (задача 3): 12](#_Toc200808740)

[11 Решение (задача 4) 13](#_Toc200808741)

[12 Тестирование (задача 4) 14](#_Toc200808742)

[13 Программа на C# (задача 4): 15](#_Toc200808743)

[14 Решение (задача 5) 17](#_Toc200808744)

[15 Тестирование (задача 5) 18](#_Toc200808745)

[16 Программа на C# (задача 5): 19](#_Toc200808746)

[17 Решение (задача 6) 21](#_Toc200808747)

[18 Тестирование (задача 6) 22](#_Toc200808748)

[19 Программа на C# (задача 6): 23](#_Toc200808749)

[20 Вывод 25](#_Toc200808750)

# Задание

Задача 1.

Вычислить значения функции z=sqrt((xi+ai)/2), если xi и ai - элементы массивов, состоящих из десяти элементов каждый.

Задача 2.

Найти наименьший четный среди положительных элементов массива А(10).

Задача 3.

Дана матрица M(5,5). Среди элементов побочной диагонали найти наименьший элемент.

Задача 4.

Дана матрица М(5,5). Среди элементов главной диагонали найти наибольший элемент. Вывести его на печать и обнулить строку и столбец, в котором он расположен.

Задача 5.

В матрице T(5,5) найти максимальный элемент, расположенный выше побочной диагонали, вывести его на печать, а столбец, в котором он расположен, обнулить.

Задача 6.

Дан массив целых чисел C(17) и матрица А(4,4). Из этих массивов выбрать элементы xi удовлетворяющие условию z< xi < y ( где z -это среднее арифметическое элементов массива С, а y задается с клавиатуры) сформировать из них массив X . Массив X отсортировать по убыванию.

# Решение (задача 1)

Схема алгоритма решения задания представлена на рисунке 1.

# 

Рисунок 1 – Схема алгоритма решения задания

# Тестирование (задача 1)

Примеры выполнения программы представлены на рисунке 2, 3 и 4.

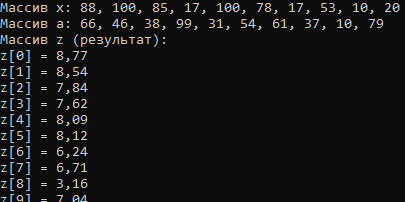


Рисунок 2 – Экранная форма программы с исходными данными x = [88, 100, 85, 17, 100, 78, 17, 53, 10, 20], a = [66, 46, 38, 99, 31, 54, 61, 37, 10, 79]

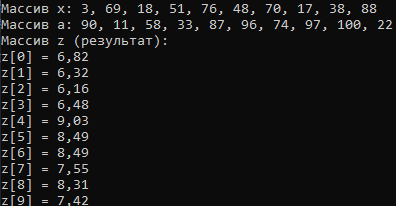


Рисунок 3 – Экранная форма программы с исходными данными x = [3, 69, 18, 51, 76, 48, 70, 17, 38, 88], a = [90, 11, 58, 33, 87, 96, 74, 97, 100, 22]

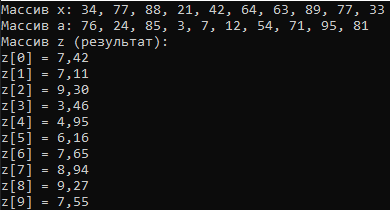


Рисунок 4 – Экранная форма программы с исходными данными x = [34, 77, 88, 21, 42, 64, 63, 89, 77, 33], a = [76, 24, 85, 3, 7, 12, 54, 71, 95, 81]

# Программа на C# (задача 1):

class Program

{

static void Main()

{

Random rand = new Random();

double[] x = new double[10];

double[] a = new double[10];

double[] z = new double[10];

for (int i = 0; i < 10; i++)

{

x[i] = rand.Next(1, 101);

a[i] = rand.Next(1, 101);

}

for (int i = 0; i < 10; i++)

{

z[i] = Math.Sqrt((x[i] + a[i]) / 2.0);

}

Console.WriteLine("Массив x: " + string.Join(", ", x));

Console.WriteLine("Массив a: " + string.Join(", ", a));

Console.WriteLine("Массив z (результат):");

for (int i = 0; i < 10; i++)

{

Console.WriteLine($"z[{i}] = {z[i]:F2}");

}

}

}

# Решение (задача 2)

Схема алгоритма решения задания представлена на рисунке 3.

# 

Рисунок 5 – Схема алгоритма решения задания

# Тестирование (задача 2)

Примеры выполнения программы представлены на рисунках 6, 7 и 8.



Рисунок 6 – Экранная форма программы с исходными данными A = [-46, 49, 81, -47, -1, 19, 3, 70, 29, -14]



Рисунок 7 – Экранная форма программы с исходными данными A = [-47, 39, 39, 47, 40, 58, -14, 94, -30, 82]



Рисунок 8 – Экранная форма программы с исходными данными A = [-23, 91, -6, -46, 61, 13, -5, 90, -16, -29]

# Программа на C# (задача 2):

class Program

{

static void Main()

{

Random rand = new Random();

int[] A = new int[10];

for (int i = 0; i < 10; i++)

{

A[i] = rand.Next(-50, 100);

}

int? minEven = null;

foreach (int num in A)

{

if (num > 0 && num % 2 == 0)

{

if (minEven == null || num < minEven)

{

minEven = num;

}

}

}

Console.WriteLine("Массив A: " + string.Join(", ", A));

if (minEven != null)

{

Console.WriteLine($"Наименьший чётный положительный элемент: {minEven}");

}

else

{

Console.WriteLine("Нет положительных чётных элементов в массиве.");

}

}

}

# Решение (задача 3)

Схема алгоритма решения задания представлена на рисунке 9

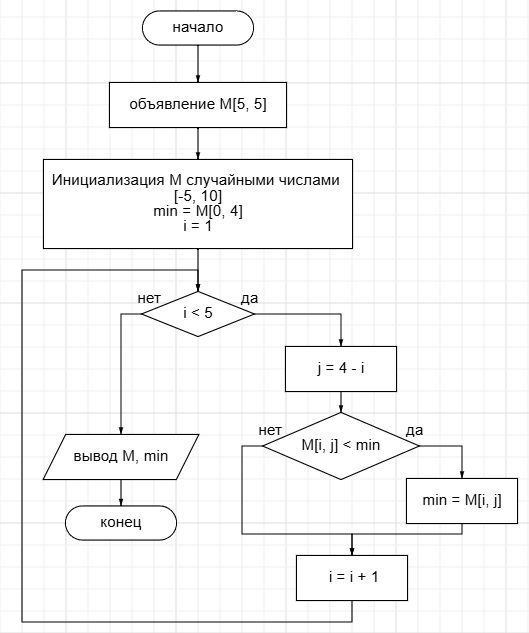


Рисунок 9 – Схема алгоритма решения задания

# Тестирование (задача 3)

Примеры выполнения программы представлены на рисунках 10, 11 и 12.

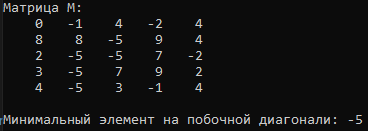


Рисунок 10 – Экранная форма программы, M = [[ 0, -1, 4, -2, 4], [ 8, 8, -5, 9, 4], [ 2, -5, -5, 7, -2], [ 3, -5, 7, 9, 2], [ 4, -5, 3, -1, 4]]

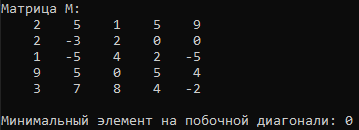


Рисунок 11 – Экранная форма программы, M = [[ 2, 5, 1, 5, 9], [ 2, -3, 2, 0, 0],[ 1, -5, 4, 2, -5], [ 9, 5, 0, 5, 4], [ 3, 7, 8, 4, -2]]

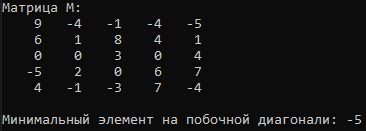


Рисунок 12 – Экранная форма программы, M = [[ 9, -4, -1, -4, -5], [ 6, 1, 8, 4, 1], [ 0, 0, 3, 0, 4], [-5, 2, 0, 6, 7], [ 4, -1, -3, 7, -4]]

# Программа на C# (задача 3):

class Program

{

static void Main()

{

int[,] M = new int[5, 5];

Random rand = new Random();

for (int i = 0; i < 5; i++)

{

for (int j = 0; j < 5; j++)

{

M[i, j] = rand.Next(-5, 10);

}

}

int min = M[0, 4];

for (int i = 1; i < 5; i++)

{

int j = 4 - i;

if (M[i, j] < min)

{

min = M[i, j];

}

}

Console.WriteLine("Матрица M:");

for (int i = 0; i < 5; i++)

{

for (int j = 0; j < 5; j++)

{

Console.Write($"{M[i, j],5}");

}

Console.WriteLine();

}

Console.WriteLine($"\nМинимальный элемент на побочной диагонали: {min}");

}

}

# Решение (задача 4)

Схема алгоритма решения задания представлена на рисунке 13

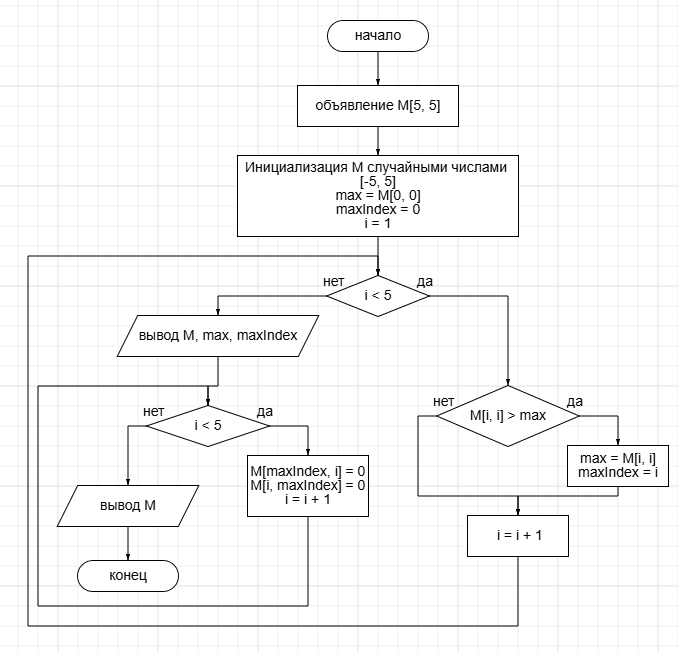
. 

Рисунок 13 – Схема алгоритма решения задания

# Тестирование (задача 4)

Примеры выполнения программы представлены на рисунках 14, 15 и 16.

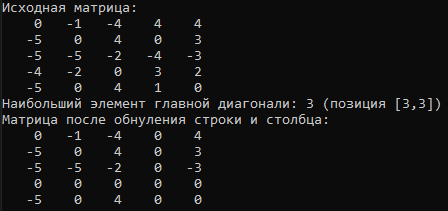


Рисунок 14 – Экранная форма программы, M = [[ 0, -1, -4, 4, 4], [-5, 0, 4, 0, 3], [-5, -5, -2, -4, -3], [-4, -2, 0, 3, 2], [-5, 0, 4, 1, 0]]

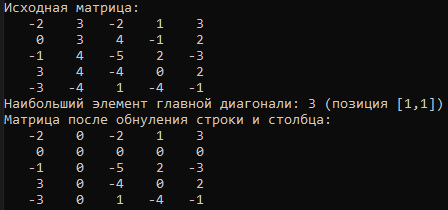


Рисунок 15 – Экранная форма программы, M = [[-2, 3, -2, 1, 3], [ 0, 3, 4, -1, 2], [-1, 4, -5, 2, -3], [ 3, 4, -4, 0, 2], [-3, -4, 1, -4, -1]]

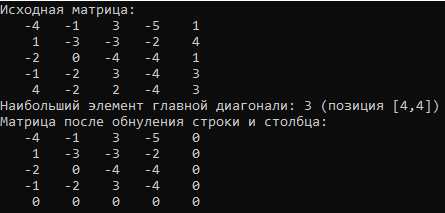


Рисунок 16 – Экранная форма программы, M = [[-4, -1, 3, -5, 1], [ 1, -3, -3, -2, 4], [-2, 0, -4, -4, 1], [-1, -2, 3, -4, 3], [ 4, -2, 2, -4, 3]]

# Программа на C# (задача 4):

class Program

{

static void Main()

{

int[,] M = new int[5, 5];

Random rand = new Random();

for (int i = 0; i < 5; i++)

{

for (int j = 0; j < 5; j++)

{

M[i, j] = rand.Next(-5, 5);

}

}

Console.WriteLine("Исходная матрица:");

PrintMatrix(M);

int max = M[0, 0];

int maxIndex = 0;

for (int i = 1; i < 5; i++)

{

if (M[i, i] > max)

{

max = M[i, i];

maxIndex = i;

}

}

Console.WriteLine($"Наибольший элемент главной диагонали: {max} (позиция [{maxIndex},{maxIndex}])");

for (int i = 0; i < 5; i++)

{

M[maxIndex, i] = 0;

M[i, maxIndex] = 0;

}

Console.WriteLine("Матрица после обнуления строки и столбца:");

PrintMatrix(M);

}

static void PrintMatrix(int[,] matrix)

{

for (int i = 0; i < 5; i++)

{

for (int j = 0; j < 5; j++)

{

Console.Write($"{matrix[i, j],5}");

}

Console.WriteLine();

}

}

}

# Решение (задача 5)

Схема алгоритма решения задания представлена на рисунке 17.

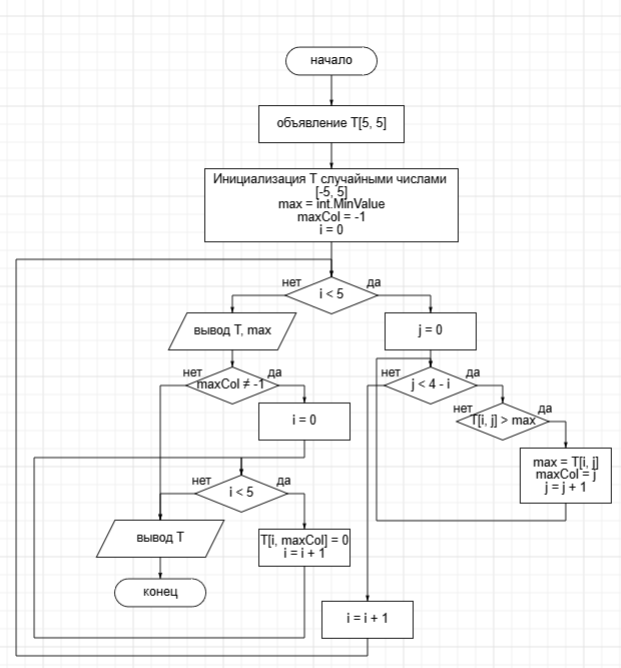


Рисунок 17 – Схема алгоритма решения задания

# Тестирование (задача 5)

Примеры выполнения программы представлены на рисунках 18, 19 и 20.

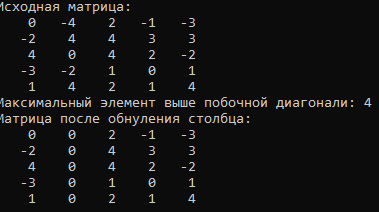


Рисунок 18 – Экранная форма программы, T = [[0, -4, 2, -1, -3], [-2, 4, 4, 3, 3], [4, 0, 4, 2, -2], [-3, -2, 1, 0, 1], [1, 4, 2, 1, 4]]

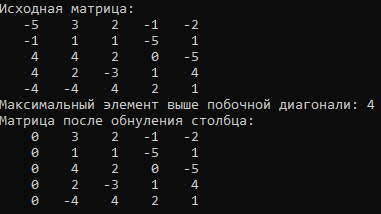


Рисунок 19 – Экранная форма программы, T = [[-5, 3, 2, -1, -2], [-1, 1, 1, -5, 1], [4, 4, 2, 0, -5], [4, 2, -3, 1, 4], [-4, -4, 4, 2, 1]]

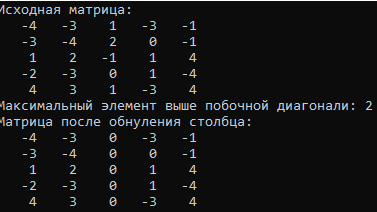


Рисунок 20 – Экранная форма программы, T = [[-4, -3, 1, -3, -1], [-3, -4, 2, 0, -1], [1, 2, -1, 1, 4], [-2, -3, 0, 1, -4], [4, 3, 1, -3, 4]]

# Программа на C# (задача 5):

class Program

{

static void Main()

{

int[,] T = new int[5, 5];

Random rand = new Random();

for (int i = 0; i < 5; i++)

for (int j = 0; j < 5; j++)

T[i, j] = rand.Next(-5, 5);

int max = int.MinValue;

int maxCol = -1;

for (int i = 0; i < 5; i++)

{

for (int j = 0; j < 5 - 1 - i; j++)

{

if (T[i, j] > max)

{

max = T[i, j];

maxCol = j;

}

}

}

Console.WriteLine("Исходная матрица:");

PrintMatrix(T);

Console.WriteLine($"Максимальный элемент выше побочной диагонали: {max}");

if (maxCol != -1)

{

for (int i = 0; i < 5; i++)

T[i, maxCol] = 0;

}

Console.WriteLine("Матрица после обнуления столбца:");

PrintMatrix(T);

}

static void PrintMatrix(int[,] matrix)

{

for (int i = 0; i < 5; i++)

{

for (int j = 0; j < 5; j++)

Console.Write($"{matrix[i, j],5}");

Console.WriteLine();

}

}

}

# Решение (задача 6)

Схема алгоритма решения задания представлена на рисунке 21.

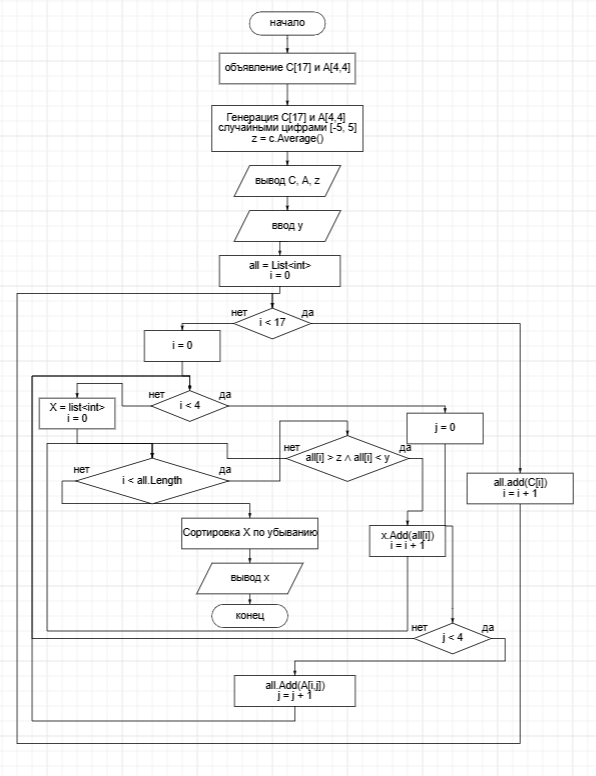


Рисунок 21 – Схема алгоритма решения задания

# Тестирование (задача 6)

Примеры выполнения программы представлены на рисунках 22, 23 и 24.

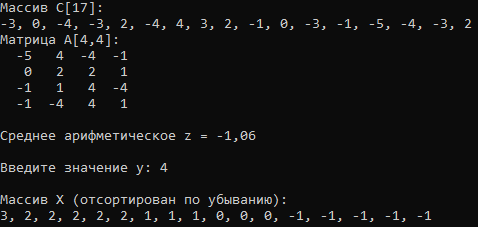


Рисунок 22 – Экранная форма программы

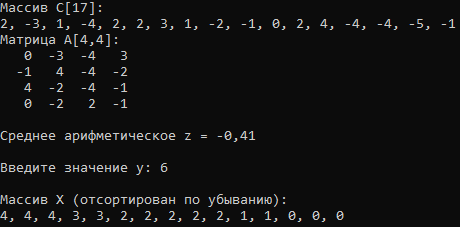


Рисунок 23 – Экранная форма программы

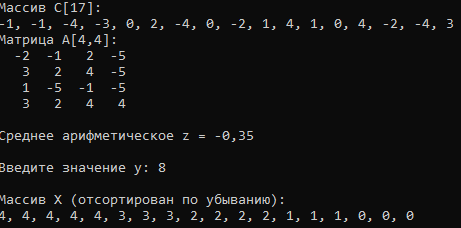


Рисунок 24 – Экранная форма программы

# Программа на C# (задача 6):

using System;

using System.Collections.Generic;

using System.Linq;

class Program

{

static void Main()

{

Random rand = new Random();

int[] C = new int[17];

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

{

C[i] = rand.Next(-5, 5);

}

int[,] A = new int[4, 4];

for (int i = 0; i < 4; i++)

for (int j = 0; j < 4; j++)

A[i, j] = rand.Next(-5, 5);

Console.WriteLine("Массив C[17]:");

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

Console.WriteLine("Матрица A[4,4]:");

for (int i = 0; i < 4; i++)

{

for (int j = 0; j < 4; j++)

Console.Write($"{A[i, j],4}");

Console.WriteLine();

}

double z = C.Average();

Console.WriteLine($"\nСреднее арифметическое z = {z:F2}");

Console.Write("\nВведите значение y: ");

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

List<int> allElements = new List<int>(C);

for (int i = 0; i < 4; i++)

for (int j = 0; j < 4; j++)

allElements.Add(A[i, j]);

List<int> X = allElements

.Where(xi => xi > z && xi < y)

.OrderByDescending(xi => xi)

.ToList();

Console.WriteLine("\nМассив X (отсортирован по убыванию):");

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

}

}

# Вывод

В ходе выполнения этих задач я освоил работу с одномерными и двумерными массивами, научился использовать циклы для перебора элементов, условные операторы для фильтрации данных, а также методы сортировки и манипуляции списками. Кроме того, я понял, как структурировать алгоритмы через блок-схемы, что помогает визуализировать логику программ и упрощает анализ и отладку