**Лабораторная работа № 5**

Цель работы:

* Разобраться с классом Random
* Решение различных задач с использованием одномерных и двумерных массивов

**1. Практическая работа**

**Задание 11.** Программа, содержащая 4 метода: создание и вывод случайного числа до 10, нахождение максимума, минимума и суммы 3 чисел. Модифицируйте программу, добавив метод нахождения суммы квадратов 3 целых чисел.

**Код программы:**

using System;

namespace Lab5

{

class Program

{

static void Main(string[] args)

{

ThreeInt chisla = new ThreeInt();

chisla.Vyvod();

Console.WriteLine("Максимальное число {0}", chisla.MaxDig());

Console.WriteLine("Минимальное число {0}", chisla.MinDig());

Console.WriteLine("Сумма чисел " + chisla.Summa());

Console.WriteLine("Сумма квадратов чисел " + chisla.SummaQvadratov());

Console.ReadLine();

}

}

}

**Класс ThreeInt:**

using System;

using System.Collections.Generic;

using System.Text;

namespace Lab5

{

class ThreeInt

{

// поля класса

private int d1, d2, d3;

/// <summary>

/// Конструктор без параметров: создаются 3 случайных числа до 10

/// </summary>

public ThreeInt()

{

// демонстрация класса Random

const int maxd = 10;

Random rnd = new Random(); // первоначально создается объект случ величины

d1 = rnd.Next(maxd);

d2 = rnd.Next(maxd);

d3 = rnd.Next(maxd);

}

/// <summary>

/// Вывод всех чисел на экран

/// </summary>

public void Vyvod() //При использовании в качестве типа возвращаемого значения для метода,void указывает, что метод не возвращает значение

{

Console.WriteLine("Первое число {0}",d1);

Console.WriteLine("Второе число {0}",d2);

Console.WriteLine("Третье число {0}",d3);

}

/// <summary>

/// Поиск максимального из чисел

/// </summary>

/// <returns>Возвращает максимум</returns>

public int MaxDig()

{

int dmax;

if (d1 > d2) dmax = d1;

else dmax = d2;

if (d3 > dmax) dmax = d3;

return dmax;

}

/// <summary>

/// Поиск минимального из чисел

/// </summary>

/// <returns>Возвращает минимум</returns>

public int MinDig()

{

int dmin;

dmin = (d1 > d2) ? d2 : d1;

dmin = (d3 > dmin) ? dmin : d3;

return dmin;

}

/// <summary>

/// Расчет суммы чисел

/// </summary>

/// <returns>Возвращает сумму</returns>

public int Summa()

{

return d1 + d2 + d3;

}

public int SummaQvadratov()

{

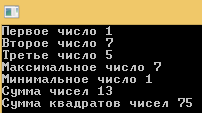
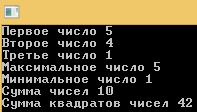
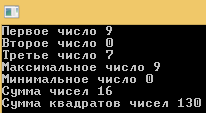
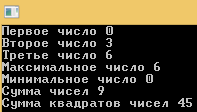
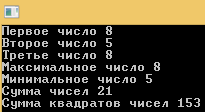
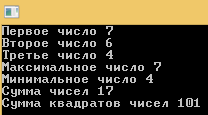
return d1\*d1 + d2\*d2 + d3\*d3;

}

}

}

**Примеры работы программы:**



**Задание 12.** Введите код программы «Решения треугольников по 3 сторонам», комментарии вводить не обязательно, запустите программу на выполнение, проверьте различные варианты выполнения программы. Модифицируйте программу, добавив код, проверяющий, является ли треугольник прямоугольным.

**Код программы:**

using System;

namespace Lab5

{

class Program

{

static void Main(string[] args)

{

try

{

Treug t = new Treug(5, 3, 4);

t.Vyvod();

Console.WriteLine("Периметр треугольника {0}", t.Perimetr());

Console.WriteLine("Площадь треугольника {0}", t.Ploshad());

if (t.IsPryamougolnu() == true)

Console.WriteLine("Прямоугольный треугольник");

else

Console.WriteLine("Это треугольник не является прямоугольным");

}

catch (ArgumentOutOfRangeException)

{

Console.WriteLine("Ошибка определения сторон треугольника");

}

}

}

}

**Класс Treug:**

using System;

using System.Collections.Generic;

using System.Text;

namespace Lab5

{

class Treug

{

// поля

private double a, b, c; // стороны треугольника

/// <summary>

/// Конструктор

/// </summary>

/// <param name="a"></param>

/// <param name="b"></param>

/// <param name="c"></param>

public Treug(double a, double b, double c)

{

if (a <= 0 || b <= 0 || c <= 0) throw (new ArgumentOutOfRangeException()); // генерация исключения

// отсортировать, чтобы a < b < c

if (a > b) { double t = a; a = b; b = t; }

if (a > c) { double t = a; a = c; c = t; }

if (b > c) { double t = b; b = c; c = t; }

// проверить соблюдения основного соотношения сторон

if (a + b <= c) throw (new ArgumentOutOfRangeException()); // генерация исключения

// поля класса отсортированы по возрастанию

this.a = a; this.b = b; this.c = c;

}

/// <summary>

/// Вывод данных

/// </summary>

public void Vyvod()

{

Console.WriteLine("a= {0}", this.a);

Console.WriteLine("b= {0}", this.b);

Console.WriteLine("c= {0}", this.c);

}

/// <summary>

/// Расчет периметра

/// </summary>

/// <returns></returns>

public double Perimetr()

{

return this.a + this.b + this.c;

}

/// <summary>

/// Расчет площади по формуле Герона

/// </summary>

/// <returns></returns>

public double Ploshad()

{

double p = Perimetr() / 2;

return Math.Sqrt(p \* (p - this.a) \* (p - this.b) \* (p - this.c));

}

public bool IsPryamougolnu()

{

if (Math.Pow(a, 2) + Math.Pow(b, 2) == Math.Pow(c, 2))

return true;

else

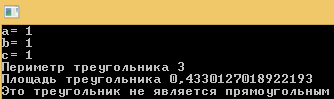
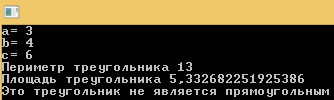
return false;

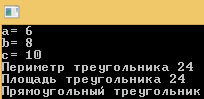
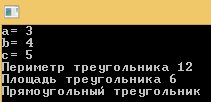
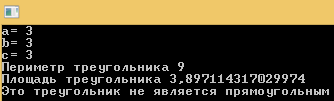
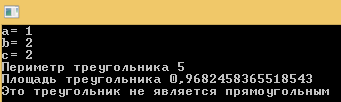
}

}

}

**Примеры работы программы:**





**Задание 13.** Введите код программы, комментарии вводить не обязательно, запустите программу на выполнение, проверьте различные варианты выполнения программы. Модифицируйте программу, добавив массивы v3=v1+v2; v4=a+c+u. Вывести на печать все массивы.

**Код программы:**

using System;

namespace Lab5

{

class Program

{

static void Main(string[] args)

{

//объявляются три одномерных массива A,B,C

int[] a, b, c;

a = new int[5];

b = new int[6];

c = new int[4];

Random rnd = new Random();

//Заполняем массив а случайными числами

for (int i = 0; i < a.GetLength(0)/\*обозначает "до конца массива а"\*/; i++)

a[i] = rnd.Next(1, 100);

//Заполняем массив b случайными числами

for (int i = 0; i < b.GetLength(0)/\*обозначает "до конца массива b"\*/; i++)

b[i] = rnd.Next(1, 100);

//Рассчитываем массив с

for (int i = 0; i < c.GetLength(0); i++)

c[i] = a[i] + b[i];

//объявление массива с явной инициализацией

int[] x = { 5, 5, 6, 6, 7, 7 };

//объявление массивов с отложенной инициализацией

int[] u, v1, v2;

u = new int[3];

for (int i = 0; i < 3; i++) u[i] = i + 1;

//v1={1,2,3};недопустимое присваивание

v1 = new int[4];

v1 = u; //допустимое присваивание

v2 = new int[4];

for (int i = 0; i < 3; i++)

v2[i] = u[i] \* 2;

int[] v3 = new int[v1.Length + v2.Length];

for(int i = 0; i < v3.Length; i++)

{

if (i < v1.Length)

v3[i] = v1[i];

else

v3[i] = v2[i - v1.Length];

}

int[] v4 = new int[a.Length + c.Length + u.Length];

for (int i = 0; i < v4.Length; i++)

{

if (i < a.Length)

v4[i] = a[i];

else if (i >= a.Length && i - a.Length < c.Length)

v4[i] = c[i - a.Length];

else

v4[i] = u[i - c.Length - a.Length];

}

// Создадим Класс Arr с методом PrintArr для вывода массивов на печать

Arr.PrintArr("a", a);

Arr.PrintArr("b", b);

Arr.PrintArr("c", c);

Arr.PrintArr("x", x);

Arr.PrintArr("u", u);

Arr.PrintArr("v1", v1);

Arr.PrintArr("v2", v2);

Arr.PrintArr("v3", v3);

Arr.PrintArr("v4", v4);

Console.ReadLine();

}

}

}

**Класс Arr:**

using System;

using System.Collections.Generic;

using System.Text;

namespace Lab5

{

class Arr

{

public static void PrintArr(string name, int[] T)

{

Console.WriteLine(name);

for (int i = 0; i < T.GetLength(0); i++)

Console.Write("\t" + name + "[{0}]={1}", i, T[i]);

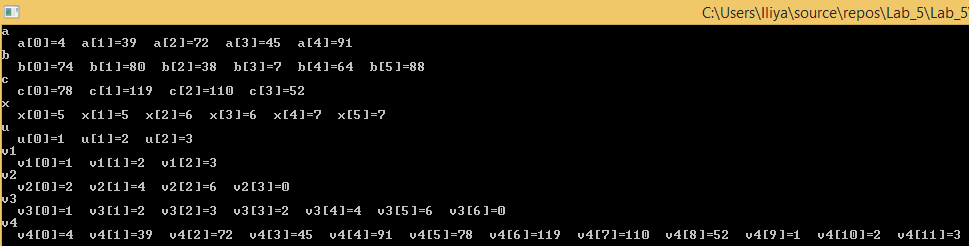
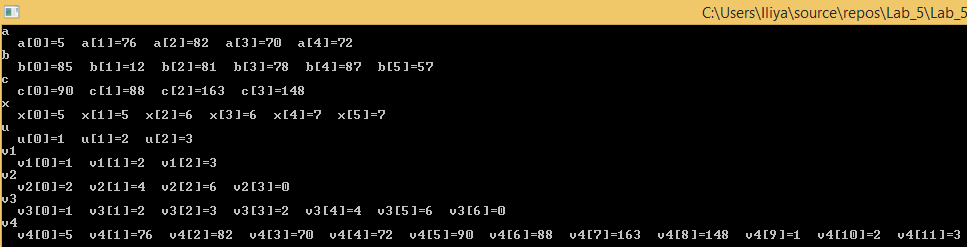
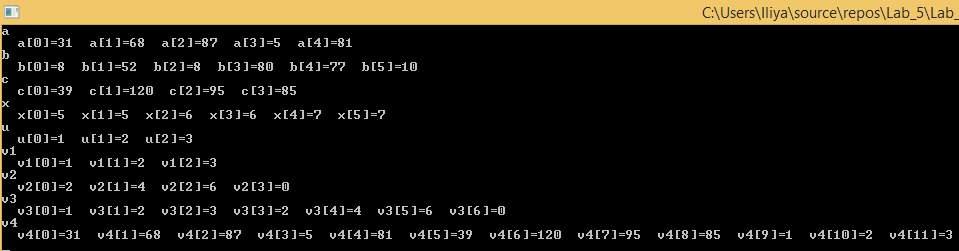
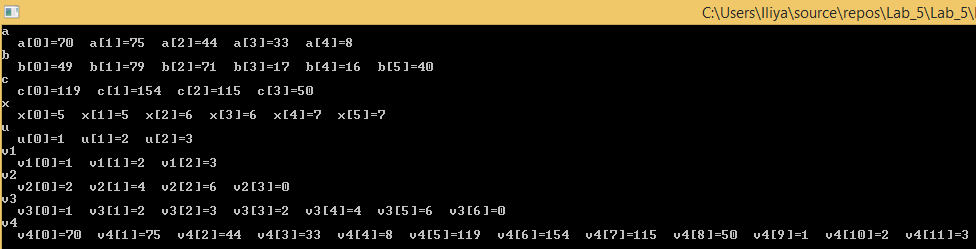
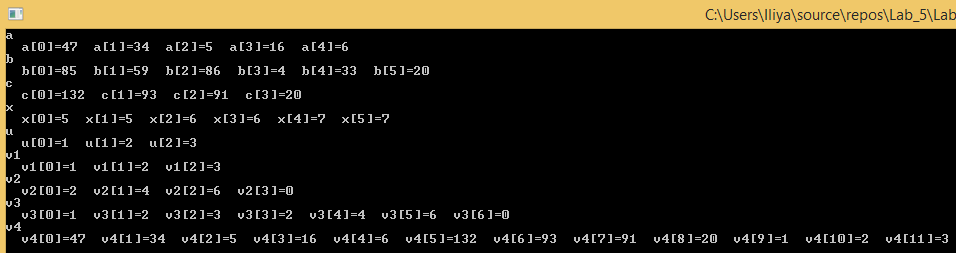
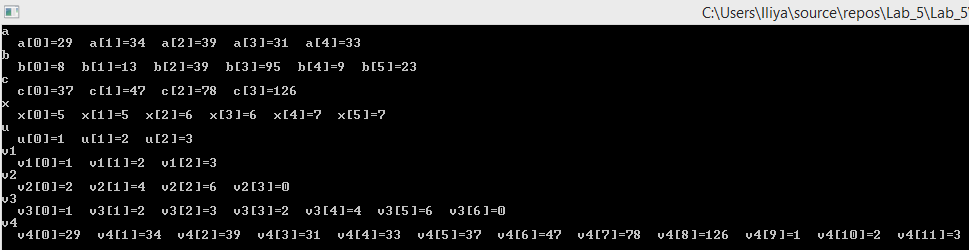
Console.WriteLine();

}

}

}

**Примеры работы программы:**



**Задание 14.** Пример программы умножения элементов матрицы на 7. Модифицируйте программу, добавив умножение элементов главной диагонали на 3.

**Код программы:**

using System;

namespace Lab5

{

class Program

{

static void Main(string[] args)

{

//объявление динамического массива v

Console.WriteLine("Введите число строк матрицы ");

int size1 = int.Parse(Console.ReadLine());

Console.WriteLine("Введите число столбцов матрицы ");

int size2 = int.Parse(Console.ReadLine());

int[,] v = new int[size1, size2];

//Заполнение матрицы случайными числами

Random rnd = new Random();

for (int i = 0; i < v.GetLength(0); i++)

for (int j = 0; j < v.GetLength(1); j++)

{

v[i, j] = rnd.Next(1, 10); ;

}

//Вывод матрицы

Console.WriteLine("Исходная матрица");

for (int i = 0; i < v.GetLength(0); i++)

{

for (int j = 0; j < v.GetLength(1); j++)

{

Console.Write(v[i, j] + "\t");

}

Console.WriteLine("\n");

}

Console.WriteLine();

//Умножение всех элементов матрицы на 7

for (int i = 0; i < v.GetLength(0); i++)

for (int j = 0; j < v.GetLength(1); j++)

{

v[i, j] = v[i, j] \* 7;

}

//Вывод новой матрицы

Console.WriteLine("Матрица\*7");

for (int i = 0; i < v.GetLength(0); i++)

{

for (int j = 0; j < v.GetLength(1); j++)

{

Console.Write(v[i, j] + "\t");

}

Console.WriteLine("\n");

}

//Умножение всех элементов главной диагонали на 3

int d = Math.Min(v.GetLength(0), v.GetLength(1));

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

v[i, i] = v[i, i] \* 3;

//Вывод новой матрицы

Console.WriteLine("Матрица, элементы главной диагонали которой \* 3");

for (int i = 0; i < v.GetLength(0); i++)

{

for (int j = 0; j < v.GetLength(1); j++)

{

Console.Write(v[i, j] + "\t");

}

Console.WriteLine("\n");

}

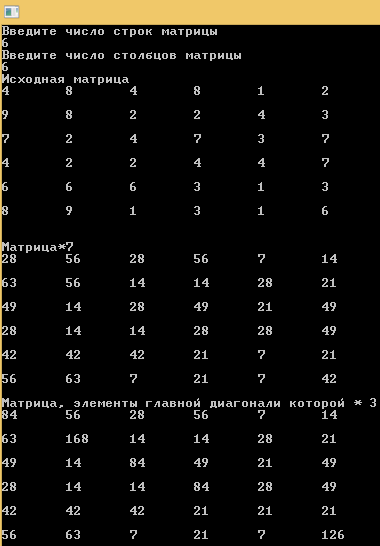
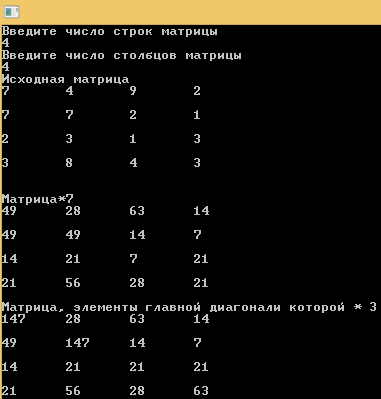
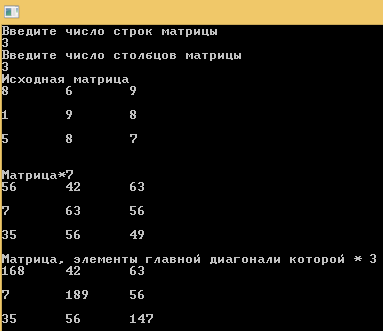
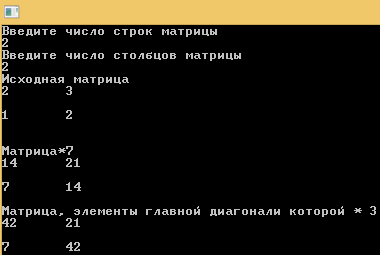
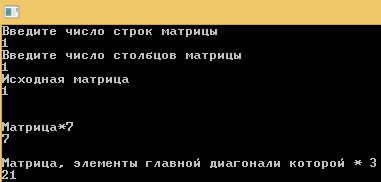
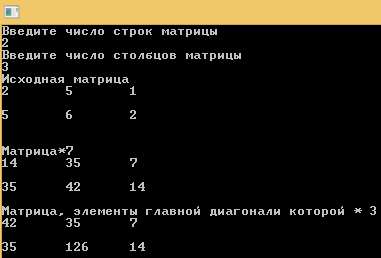
Console.ReadLine();

}

}

}

**Примеры работы программы:**



**Задание 15.** Класс квадратное уравнение. Одномерные массивы. Откройте программу вычисления корней квадратного уравнения из Лабораторной работы №4, внесите изменения для вывода корней в массив.

**Код программы:**

using System;

namespace Lab5

{

class Program

{

static void Main(string[] args)

{

double a, b, c;

Console.Write("a = ");

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

Console.Write("b = ");

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

Console.Write("c = ");

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

KvUr uravn = new KvUr(a, b, c);

Console.WriteLine("Количество корней уравнения {0}", uravn.k);

if (uravn.k > 0)

{

Console.WriteLine("Корни уравнения:");

for (int i = 0; i < uravn.k; i++)

Console.WriteLine("Корень {0} равен {1}", i + 1, uravn.x[i]);

}

}

}

}

**Класс KvUr:**

using System;

using System.Collections.Generic;

using System.Text;

namespace Lab5

{

class KvUr

{

// поля класса

private double a, b, c; // коэффициенты уравнения

public int k; // количество корней

public double[] x; // корни уравнения

// конструктор

public KvUr(double a, double b, double c)

{

if (a == 0 && b == 0 && c == 0) throw (new ArgumentNullException()); // вырожденный случай

this.a = a; this.b = b; this.c = c;

Reshenie();

}

// метод: вычисление дискриминанта

private double Discr()

{ return Math.Pow(b, 2) - 4 \* a \* c; }

// метод: вычисление количества корней

private void CountK()

{

if (this.a == 0) this.k = 0;

else

{

double d = Discr();

if (d > 0) this.k = 2;

else if (d == 0) this.k = 1;

else this.k = 0;

}

}

// метод: вычисление корней уравнения

private void Reshenie()

{

CountK();

if (this.k == 1)

{

this.x = new double[1];

this.x[0] = -b / 2 / a;

}

else

{

this.x = new double[2];

double d = Math.Sqrt(Discr());

this.x[0] = (-b + d) / 2 / a;

this.x[1] = (-b - d) / 2 / a;

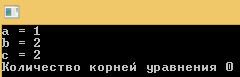
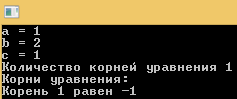
}

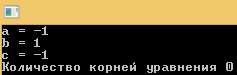
}

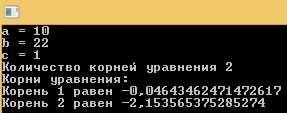
}

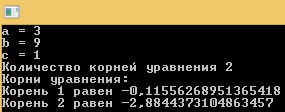
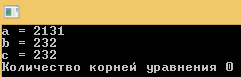
}

**Примеры работы программы:**







**Задание 16.** Введите код программы «Ввод, транспонирование и вывод матрицы»,

комментарии вводить не обязательно, запустите программу на выполнение,

проверьте различные варианты выполнения программы.

**Код программы:**

sing System;

namespace Lab5

{

class Program

{

static void Main(string[] args)

{

const int m = 3;//Размероность Матрицы задается

const int n = 4;//константами 3х4

Matrica x = new Matrica(m, n); //определение Класса Matrica с передачей размеров матрицы

Random r = new Random();//r - переменная для создания случайного числа

// заполнить случайными числами от 0 до 1

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

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

x.a[i, j] = r.NextDouble();

Console.WriteLine("\nИсходная матрица ");

x.Vyvod();//Метод вывода для матрицы X

Console.WriteLine("\nТранспонированная матрица ");

Matrica y = Matrica.Transpon(x); //метод транспонирования

y.Vyvod();//Метод выовда для матрицы Y

}

}

}

**Класс Матрица:**

sing System;

using System.Collections.Generic;

using System.Text;

namespace Lab5

{

class Matrica

{

// поля

public double[,] a; // двумерный массив действительных чисел

/// <summary>

/// Конструктор

/// </summary>

/// <param name="m"></param>

/// <param name="n"></param>

public Matrica(int m, int n)

{

if (m <= 0 || n <= 0) throw (new ArgumentOutOfRangeException());

this.a = new double[m, n]; // Выделить память для массива

}

/// <summary>

/// Вывод матрицы по строкам

/// </summary>

public void Vyvod()

{

for (int i = 0; i < this.a.GetLength(0); i++) // ф-ция GetLength() возвращает кол-во элементов в данной размерности(аргумент)

{

for (int j = 0; j < this.a.GetLength(1); j++)

Console.Write("{0,8:f4}", this.a[i, j]); // 8 - ширина поля вывода, f-fixed point, 4 - число знаков после точки

Console.WriteLine();

}

}

/// <summary>

/// Транспонирование матрицы

/// </summary>

/// <param name="mass">mass - транспонируемая матрица</param>

/// <returns></returns>

public static Matrica Transpon(Matrica mass)

{

int m = mass.a.GetLength(0);

int n = mass.a.GetLength(1);

Matrica b = new Matrica(n, m); // число строк = числу столбцов исх массива. И наоборот

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

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

b.a[i, j] = mass.a[j, i];

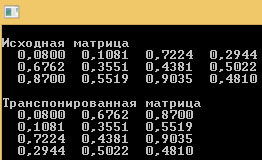
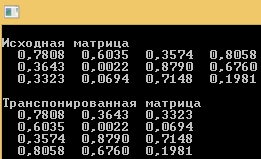
return b;

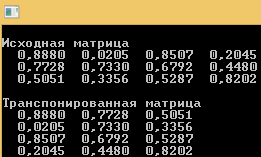
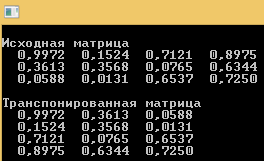
}

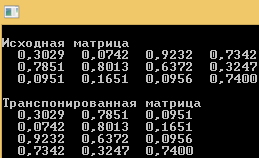
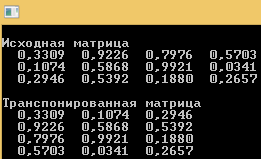
}

}

**Примеры работы программы:**

**2. Самостоятельная работа**

**2.1. Одномерные массивы**

**Вариант 17**

using System;

public class Program

{

public static void Main()

{

//Поменять местами максимальный и минимальный элемент массива

Random r = new Random();

int N, i, S;

N = r.Next(5, 10);

int[] A = new int[N];

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

{

A[i] = r.Next(0, 50);

Console.Write(A[i] + " ");

}

int min;

int max;

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

{

Console.WriteLine();

}

max = A[0];

min = A[0];

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

{

if (A[i] > max)

{

max = A[i];

}

if (A[i] < min)

{

min = A[i];

S = max; max = min; min = S;

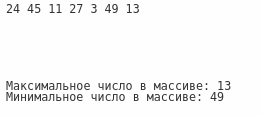
}

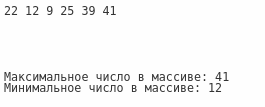
}

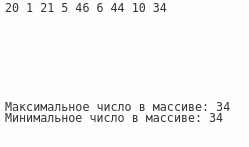
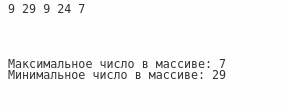
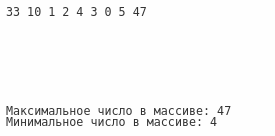
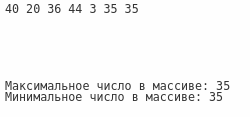
Console.WriteLine("Максимальное число в массиве: {0}", max);

Console.WriteLine("Минимальное число в массиве: {0}", min);

}

}





**Вывод:** программа самостоятельно вводит 5 рандомных чисел и определяет наибольшее и наименьшее число. Благодаря ей, у нас меньше затрачивается сил на ее решение.

**Вариант 14**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

public class Program

{

public static void Main()

{

//Найти сумму максимальных элементов каждой строки массива

double[,] a = new double[3, 4];

Random r = new Random();

for(int i =0;i<3;i++)

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

{

a[i, j] = r.NextDouble();

}

double currMax = 0;

double sum = 0;

for (int i = 0; i < 3; i++)

{

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

{

Console.Write(String.Format("{0:f} ", a[i, j]));

}

Console.WriteLine();

}

for (int i = 0; i < 3; i++)

{

currMax = a[i, 0];

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

{

if (a[i, j] > currMax) currMax = a[i, j];

}

sum += currMax;

}

Console.WriteLine("{0:f}",sum);

/\*

int[] a, b, c;

a = new int[5];

b = new int[6];

c = new int[4];

Random rnd = new Random();//Заполняеммассив а случайными числами

for (int i = 0; i < a.GetLength(0); i++)

a[i] = rnd.Next(1, 100);//Заполняем массив b случайными числами

for(int i= 0; i< b.GetLength(0); i++)

b[i] = rnd.Next(1, 100);//Рассчитываеммассивс

for (int i = 0; i < c.GetLength(0); i++)

c[i] = a[i] + b[i];//объявление массива с явной инициализацией

int[] x = {5, 5, 6, 6, 7, 7};//объявление массивов с отложенной инициали-зацией

int[] u, v1,v2;

u = new int[3];

for (int i = 0; i < 3; i++)

u[i] = i + 1;//v1={1,2,3};недопустимое присваивание

v1 = new int[4];

v1 = u; //допустимое присваивание

v2 = new int[4];

for (int i = 0; i < 3; i++)

v2[i] = u[i]\*2;// Создадим Класс Arr с методом PrintArr для вы-вода массивов на печать

Arr.PrintArr("a", a);

Arr.PrintArr("b", b);

Arr.PrintArr("c", c);

Arr.PrintArr("x", x);

Arr.PrintArr("u", u);

Arr.PrintArr("v1", v1);

Arr.PrintArr("v2", v2);

Console.ReadLine();}}

class Arr

{

public static void PrintArr(string name, int[] T)

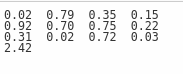
{

Console.WriteLine(name);

for (int i = 0; i < T.GetLength(0); i++)

Console.Write("\t" + name + "[{0}]={1}", i, T[i]);

Console.WriteLine();}}\*/

 }}





**Вывод:** данная программа выводит наибольшие числа каждой строчки и скадывает их. Благодаря массу можно гораздо упростить жизнь.

**2.2. Двумерные массивы**

**Вариант 12.**

using System;

public class Program

{

public static void Main()

{

//Сожмите массив, выбросив каждый второй его элемент (дополнительные массивы использовать не разрешается

Console.Write("Размер массива: ");

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

int[] a = new int[n];

Random rnd = new Random();

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

{

a[i] = rnd.Next(-10, 11);

Console.Write("" + a[i] + " ");

}

Console.WriteLine();

for (int i = 1; i < a.Length; i++)

{

for (int j = i + 1; j < a.Length; j++)

a[j - 1] = a[j];

Array.Resize(ref a, a.Length - 1);

}

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

Console.Write("" + a[i] + " ");

// Console.WriteLine();

Console.WriteLine(true);

}

}







**Вывод:** узнали о таких функциях, как сжатие массивов. Усовершеннствовали свои знания и дополнили их.

**Вариант 22.**

using System;

public class Program

{

public static void Main()

{

//Сформируйте двумерный массив NxN по сле-дующему правилу: элементы главной диагона-ли равны 1, ниже главной диагонали -0, а выше -сумме индексов.

Console.WriteLine("Введите размер квадратной матрицы");

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

int[,] a = new int[n, n];

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

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

a[i, j] = i + j;

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

for (int j = i; j <= i; j++)

a[i, j] = 1;

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

{

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

Console.Write("{0}{1}", a[i, j], "\t");

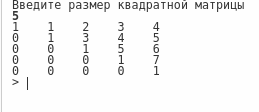
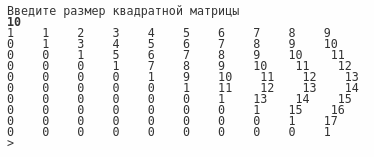
Console.WriteLine();

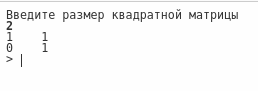
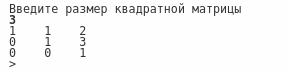
}

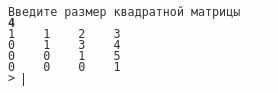
Console.ReadLine();

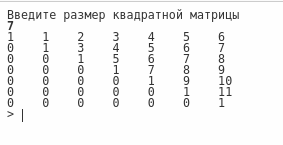
}

}









**Вывод:** благодаря двумерному массиву научились делать так, что бы элементы главной линии были равны 1, а все элементы ниже равны нулю. При том, что будет дана лишь квадратная матрица, любой размерности.