**5 Массивы. Класс Array**

Задание 1. Найти количество отрицательных элементов массива А.

Листинг программы:

namespace zad1

{

class Program

{

static int[] Input()

{

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

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

int[] a = new int[A];

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

{

Console.Write("A[{0}]= ", i);

a[i] = int.Parse(Console.ReadLine());

}

return a;

}

static void Print(int[] a)

{

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

{

Console.Write("{0} ", a[i]);

}

}

static void Change(int[] a)

{

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

{

if (a[i] < 0)

{

Console.WriteLine(a[i]);

}

}

}

static void Main()

{

int[] MyArray = Input();

Console.WriteLine("Исходный массив: ");

Print(MyArray);

Console.WriteLine("Значение меньше нуля: ");

Change(MyArray);

Console.ReadLine();

}

}

}

Таблица 5.1 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Размерность = 3  1, -2, 3 | -2 |

Анализ результатов:

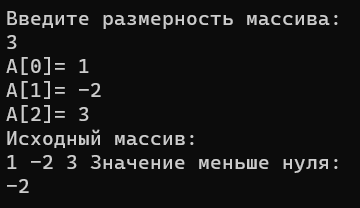


Рисунок 5.1 – Результат работы программы

Источник: собственная разработка

Задание 2. Во всех задачах сформировать и вывести на экран одномерный массив согласно варианту. Произвести его сортировку и бинарный поиск числа k (k - вводится с клавиатуры). Напишите программу для решения задачи:

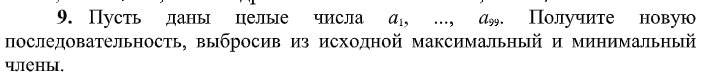


Рисунок 5.2 – Условие задания

Источник: собственная разработка

Листинг программы:

namespace zad2

{

internal class Program

{

static void Main(string[] args)

{

int[] firstArr = new int[98];

Random rand = new Random();

Console.Write("Последовательность: ");

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

{

firstArr[i] = rand.Next(10);

Console.Write($"{firstArr[i]} ");

}

Console.WriteLine();

int min = firstArr.Min();

int max = firstArr.Max();

Console.Write("Преобразованная последовательность: ");

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

{

if ((firstArr[i] != max) && (firstArr[i] != min))

{

Console.Write($"{firstArr[i]} ");

}

}

Console.ReadLine();

}

}

}

Таблица 5.2 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Последовательность от а1….а99 | Измененная последовательность |

Анализ результатов:

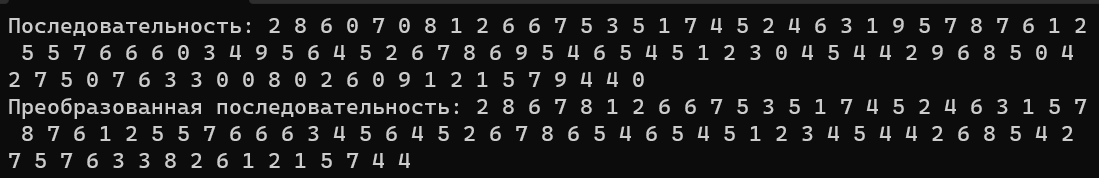


Рисунок 5.3– Результат работы программы

Источник: собственная разработка

Задание 3. В каждой задаче задается квадратная целочисленная матрица NxN (значение N вводится с клавиатуры, N<10). Программа должна заполнять матрицу случайными числами из диапазона [a, b] (a, b вводятся с клавиатуры) и осуществлять вывод на экран исходной матрицы. Затем необходимо произвести необходимые действия и напечатать результаты.

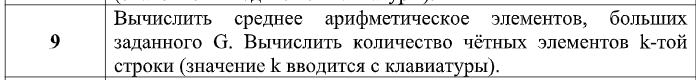


Рисунок 5.4 – Условие задания

Источник: собственная разработка

Листинг программы:

namespace zad3

{

internal class Program

{

static void Main(string[] args)

{

Console.Write("Введите размер квадратной матрицы: ");

byte N = Convert.ToByte(Console.ReadLine());

int[,] Matrix = new int[N, N];

Console.Write("Введите A: ");

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

Console.Write("Введите B: ");

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

Random random = new Random();

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

{

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

{

Matrix[i, j] = random.Next(A, B);

}

}

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

{

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

{

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

}

Console.WriteLine();

}

Console.Write("Введите G: ");

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

int GCount = 0;

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

{

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

{

if (Matrix[i, j] > G)

{

GCount++;

}

}

}

Console.WriteLine($"Количество элементов больше G: {GCount}");

Console.Write("Введите строку: ");

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

int odd = 0;

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

{

if (Matrix[k, j] % 2 == 0)

{

odd++;

}

}

Console.WriteLine($"Количество четных элементов {k} строки: {odd}");

Console.ReadLine();

}

}

Таблица 5.3 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Размер = 4, А = 1, В = 10, G = 4 | Матрица, Кол-во элементов больше G, Кол-во четных элементов. |

Анализ результатов:

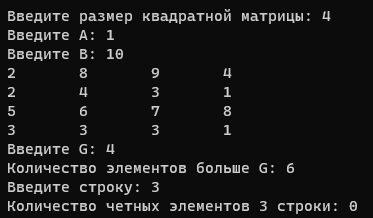


Рисунок 5.5 – Результат работы программы

Источник: собственная разработка

Задание 4. Решить задачу на проверку условий после выполнения расчетов:



Рисунок 5.6 – Условие задания

Источник: собственная разработка

Листинг программы:

namespace zad4

{

internal class Program

{

static void Main(string[] args)

{

int[,] Arr = new int[3, 5];

Random random = new Random();

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

{

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

{

Arr[i, j] = random.Next(0, 10);

}

}

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

{

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

{

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

}

Console.WriteLine();

}

int firstSum = 0;

int secondSum = 0;

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

{

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

{

if (i == 0)

{

firstSum += Arr[i, j];

}

else if (i == Arr.GetLength(0) - 2)

{

secondSum += Arr[i, j];

}

}

}

if (firstSum > secondSum)

Console.WriteLine($"Сумма элементов первой строки больше {firstSum} > {secondSum}");

else if (firstSum < secondSum)

Console.WriteLine($"Сумма элементов предпоследнего столбца больше {secondSum} > {firstSum}");

else

Console.WriteLine("Столбцы равны");

Console.ReadLine();

}

}

}

Таблица 5.4 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Матрица | Сумма элементов первой строки |

Анализ результатов:

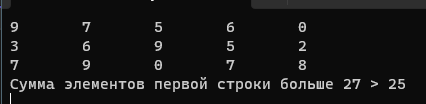


Рисунок 5.7 – Результат работы программы

Источник: собственная разработка

Задание 5. Разработать рекурсивный алгоритм и программу решения задачи, в которой вычислить: f(n)=(n+2)!/(n+4)!. Исходные данные вводятся с клавиатуры.

Листинг программы:

namespace zad5

{

internal class Program

{

public static double Fact(int x)

{

double factorial = 1;

while (x > 1)

{

factorial \*= x;

x--;

}

return factorial;

}

static void Main(string[] args)

{

Console.Write("Введите n: ");

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

double ans = Fact(n + 2) / Fact(n + 4);

Console.WriteLine($"f({n}) = {ans}");

Console.ReadLine();

}

}

}

Таблица 5.5 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| N = 100 | 9,33 |

Анализ результатов:



Рисунок 5.8 – Результат работы программы

Источник: собственная разработка

Дополнительное задание: Заполнить двумерный массив размером 7х7 числами 1, 2, …, 49, расположенными в нем по спирали

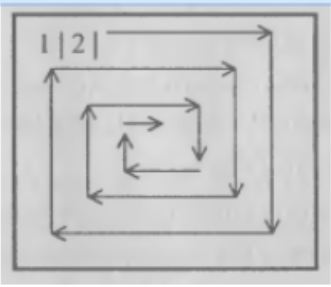


Рисунок 5.9 – Спираль

Источник: собственная разработка

Листинг программы:

namespace zad6

{

internal class Program

{

const int N = 7;

static void Main(string[] args)

{

int n = 0;

int[,] helix = new int[N, N];

int x0 = 0;

int xn = N - 1;

int y0 = 0;

int yn = N - 1;

while (n < N \* N)

{

for (int x = x0; x <= xn; x++)

{

helix[y0, x] = ++n;

}

y0++;

for (int y = y0; y <= yn; y++)

{

helix[y, xn] = ++n;

}

xn--;

for (int x = xn; x >= x0; x--)

{

helix[yn, x] = ++n;

}

yn--;

for (int y = yn; y >= y0; y--)

{

helix[y, x0] = ++n;

}

x0++;

}

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

{

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

{

Console.Write("{0}\t", helix[y, x]);

}

Console.WriteLine();

}

Console.ReadLine();

}

}

}

Таблица 5.6 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Const = 7 | Спиральная матрица |

Анализ результатов:

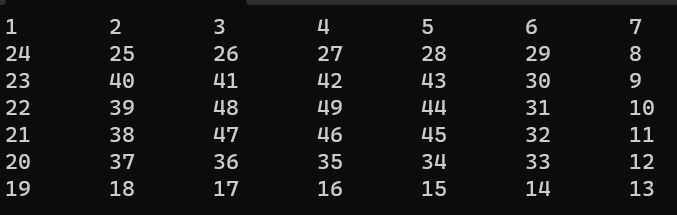


Рисунок 5.10 – Результат работы программы

Источник: собственная разработка