1. **Массивы. Класс Array**

Задание 1. Вывести на экран дисплея номера элементов массива удовлетворяющих условию 0<xi<3,2.

Листинг программы:

namespace Task\_1

{

class Program

{

public static void Main(string[] args)

{

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

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

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

double[] array = CreateArray(number);

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

{

if (GetIndex(array[i]))

{

Console.WriteLine($"Индекс: {i}");

}

}

}

public static double[] CreateArray(int number)

{

double[] array = new double[number];

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

{

array[i] = double.Parse(Console.ReadLine()); ;

}

return array;

}

public static bool GetIndex(double numX)

{

if (numX > 0 && numX < 3.2)

{

return true;

}

else

{

return false;

}

}

}

}

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 1,2 2,2 5 0,4 | 0 1 3 |

Анализ результатов:

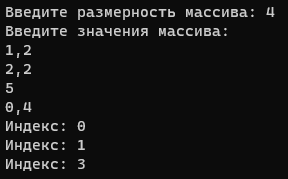


Рисунок 1.1 – Результат работы программы

Источник: собственная разработка

Задание 2. Пусть даны вещественные числа а1…а20. Преобразуйте эту последовательность по правилу: большее из а1 и а10+i (i = 1,…,10) примите в качестве нового значения аi, меньшее – в качестве нового значения а10+i.

Листинг программы:

namespace Task\_2

{

class Program

{

public static void Main(string[] args)

{

Console.Write("Введите размер массива: ");

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

Console.WriteLine("Исходный массив");

int[] array = GetCreateArray(size);

PrintArray(array);

Console.Write("\nВведите индекс (0-9) для замены: ");

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

Console.WriteLine("Массив с заменами");

PrintArray(Function(array, index));

}

public static int[] Function(int[] array, int num)

{

int max = array[num];

int min = array[num];

int size = array.Length;

for (int i = num; i < size; i++)

{

if (array[i] > max) max = array[i];

else

if (array[i] < min) min = array[i];

}

array[num] = max;

array[num + 10] = min;

return array;

}

public static int[] GetCreateArray(int size)

{

int[] array = new int[size];

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

{

array[i] = GetNumber();

}

return array;

}

public static int GetNumber()

{

Console.Write("Введмте значение: ");

return int.Parse(Console.ReadLine());

}

public static void PrintArray(int[] array)

{

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

{

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

}

}

}

}

Таблица 2.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 20  4 5 6 7 8 2 3 4 7 55 66 77 3 0 22 33 44 55 6 56  8 | 4 5 6 7 8 2 3 4 7 55 66 77 3 0 22 33 44 55 6 56  4 5 6 7 8 2 3 4 77 55 66 77 3 0 22 33 44 55 0 56 |

Анализ результатов:

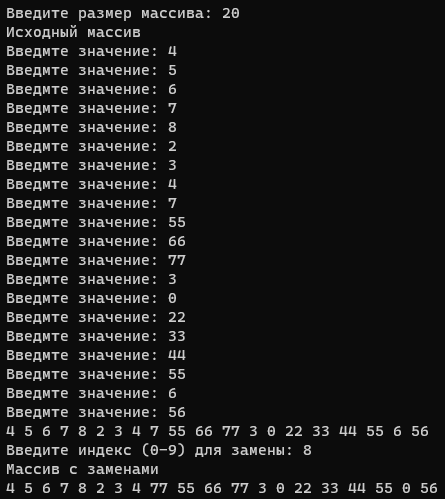


Рисунок 2.1 – Результат работы программы

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.

Вычислить сумму квадратов отрицательных чисел. Определить наименьший элемент в каждой строки.

Листинг программы:

using System;

namespace Task\_3

{

class Program

{

public static void Main(string[] args)

{

Console.Write("Введите размерность квадратной матрицы N<10: ");

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

Console.Write("Введите начальный диапазон: ");

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

Console.Write("Введите конечный диапазон: ");

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

double[,] matrix = GetCreateSquareMatrix(number, start, finish);

Console.WriteLine("Исходная матрица");

Print(matrix, number);

Console.WriteLine($"Сумму квадратов отрицательных чисел: {GetSumSquareТegativeNumbers(matrix, number)}") ;

Console.WriteLine("Наименьший элемент в каждой строки");

foreach (double element in GetMinElement(matrix, number))

{

Console.WriteLine(element);

}

}

public static double[,] GetCreateSquareMatrix(int number, int start, int finish)

{

Random rnd = new Random();

double[,] matrix = new double[number, number];

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

{

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

{

matrix[i, j] = rnd.Next(start, finish + 1);

}

}

return matrix;

}

public static int GetSumSquareТegativeNumbers(double[,] matrix, int number)

{

int sum = 0;

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

{

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

{

if (matrix[i, j] < 0)

{

sum += (int)Math.Pow(matrix[i, j], 2);

}

}

}

return sum;

}

public static double[] GetMinElement(double[,] matrix, int number)

{

double[] array = new double[number];

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

{

double min = matrix[i, 0];

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

{

if (matrix[i, j] < min)

{

min = matrix[i, j];

}

}

array[i] = min;

}

return array;

}

public static void Print(double[,] matrix, int number)

{

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

{

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

{

Console.Write(matrix[i, j] + " ");

}

Console.WriteLine();

}

}

}

}

Таблица 3.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 6 -4 10 | 57 -1 -2 -3 -4 -4 -1 |

Анализ результатов:

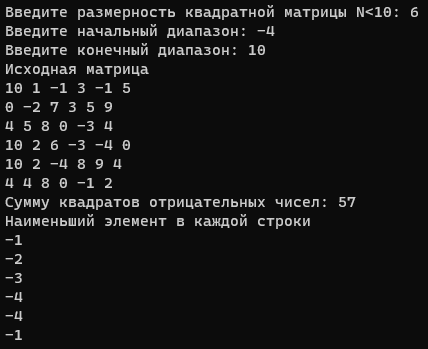


Рисунок 3.1 – Результат работы программы

Источник: собственная разработка

Задание 4. Решить задачу на проверку условий после выполнения расчетов: В двумерном массиве хранится информация о зарплате каждого из 20 сотрудников фирмы за каждый месяц года (в первом столбце – за январь, во втором – за февраль и т.д.) Верно ли, что общая зарплата всех сотрудников в феврале была меньше, чем и октябре?

Листинг программы:

namespace Task\_4

{

class Program

{

public static void Main(string[] args)

{

Console.Write("Введите кол-во сотрудников: ");

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

Console.Write("Введите кол-во месяцев: ");

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

double[,] salaryMatrix = GetCreateMatrix(people, month);

Console.WriteLine("Зарплатная матрица");

Print(salaryMatrix);

Console.Write("Введите месяц: ");

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

Console.Write("Введите месяц: ");

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

Console.WriteLine($"Зарплата всех сотрудников в {monthOne} месяце была меньше, чем в {monthTwo} месяце: {GetSalaryComparison(salaryMatrix ,monthOne, monthTwo)}");

}

public static bool GetSalaryComparison(double[,] matrix, int monthOne, int monthTwo)

{

double salarySumOne = 0;

double salarySumTwo = 0;

int rows = matrix.GetUpperBound(0) + 1; // количество строк

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

{

salarySumOne += matrix[i, monthOne - 1];

salarySumTwo += matrix[i, monthTwo - 1];

}

if (salarySumOne > salarySumTwo)

{

return true;

}

return false;

}

public static double[,] GetCreateMatrix(int row, int colum)

{

double[,] matrix = new double[row, colum];

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

{

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

{

matrix[i, j] = GetNumberRandom(); ;

}

}

return matrix;

}

public static double GetNumberRandom()

{

Random rnd = new Random();

return rnd.Next(9, 100);

}

public static double GetNumber()

{

Console.Write("Введите зарплату: ");

return double.Parse(Console.ReadLine());

}

public static void Print(double[,] matrix)

{

int rows = matrix.GetUpperBound(0) + 1; // количество строк

int columns = matrix.Length / rows;

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

{

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

{

Console.Write(matrix[i, j] + " ");

}

Console.WriteLine();

}

}

}

}

Таблица 4.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 20 12 2 10 | False |

Анализ результатов:

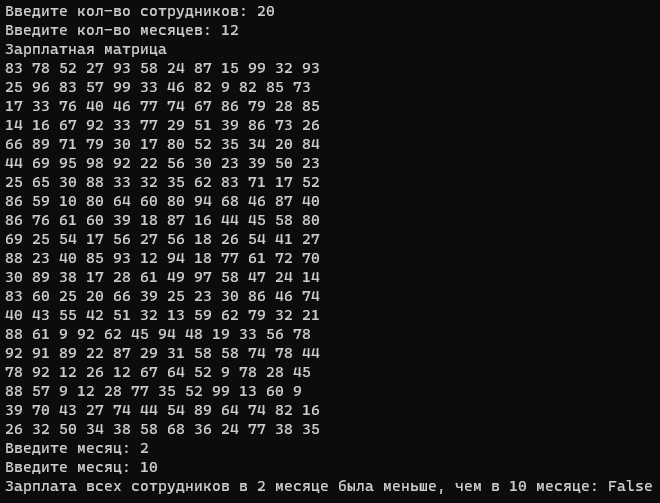


Рисунок 4.1 – Результат работы программы

Источник: собственная разработка

Задание 5. Разработать рекурсивный алгоритм и программу решения задачи, в которой вычислить: f(n)=(n-1)!/n!. Исходные данные вводятся с клавиатуры.

Листинг программы:

namespace Task\_5

{

class Program

{

public static void Main(string[] args)

{

Console.Write("Введите число для возведения в факториал: ");

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

Console.WriteLine($"Факториал({number})= {Function(number):F4}");

}

public static double Function(int number)

{

double result = Factorial(number - 1) / Factorial(number);

return result;

}

public static int Factorial(int number)

{

int result;

if (number == 0 || number == 1)

{

result = 1;

}

else

{

result = Factorial(number - 1) \* number;

}

return result;

}

}

}

Таблица 5.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 8 | 0,0000 |

Анализ результатов:



Рисунок 5.1 – Результат работы программы

Источник: собственная разработка

Задание 6. Заполнить двумерный массив размером 7х7 числами 1, 2, …, 49, расположенными в нем по спирали

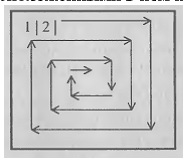


Рисунок 6.1 – Результат работы программы

Источник: собственная разработка

Листинг программы:

using System.Drawing;

namespace Task\_6

{

class Program

{

public static void Main(string[] args)

{

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

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

Print(GetCreateSquareMatrix(size), size);

}

public static double[,] GetCreateSquareMatrix(int size)

{

int number = 1;

int i = 0;

int j = 0;

int iStart = 0;

int iFinish = 0;

int jStart = 0;

int jFinish = 0;

int sizeMatrix = size \* size;

double[,] matrix = new double[size, size];

while (number <= sizeMatrix)

{

matrix[i, j] = number;

if (i == iStart && j < size - jFinish - 1)

j++;

else if (j == size - jFinish - 1 && i < size - iFinish - 1)

i++;

else if (i == size - iFinish - 1 && j > jStart)

j--;

else

i--;

if ((i == iStart + 1) && (j == jStart) && (jStart != size - jFinish - 1))

{

iStart++;

iFinish++;

jStart++;

jFinish++;

}

number++;

}

return matrix;

}

public static void Print(double[,] matrix, int size)

{

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

{

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

{

Console.Write(matrix[i, j] + " ");

}

Console.WriteLine();

}

}

}

}

Таблица 6.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 7 | Массив |

Анализ результатов:

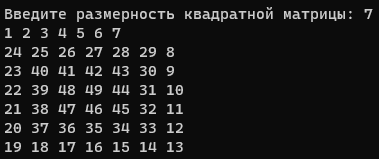


Рисунок 6.2 – Результат работы программы

Источник: собственная разработка