**4 «МАССИВЫ. АЛГОРИТМЫ ОБРАБОТКИ МАССИВОВ. РЕККУРСИЯ».**

Задание 1. Заменить четные элементы в массиве целых чисел нулями и вывести новый массив на экран.

Листинг программы:

using System;

using System.Linq;

class Program

{

static void Main()

{

Random random = new Random();

int[] arr = Enumerable.Range(1, random.Next(20)).Select(r => random.Next(1, 100)).ToArray();

Console.WriteLine("Исходный массив:");

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

arr = arr.Select((a, i) => i % 2 == 0 ? 0 : a).ToArray();

Console.WriteLine("Новый массив:");

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

Console.ReadLine();

}

}

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | Исходный массив:  59 83 41 67 7 9 19 3 98 86 6 88 60 98 24 75  Новый массив:  0 83 0 67 0 9 0 3 0 86 0 88 0 98 0 75 |

Анализ результатов:

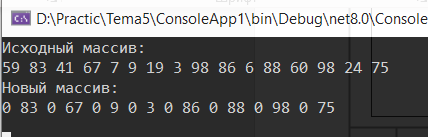


Рисунок 1.1 – Результат работы программы

Задание 2. Пусть даны целые числа а 1 , ..., а п . Если в данной последовательности ни одно четное число не расположено после нечетного, то напечатайте все отрицательные члены последовательности, иначе – все положительные. Порядок следования чисел в обоих случаях замените обратным.

Листинг программы:

using System;

class Program

{

static void Main()

{

Console.Write("Введите количество чисел в последовательности: ");

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

int[] sequence = new int[n];

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

{

Console.Write($"Введите {i + 1}-е число: ");

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

}

bool evenAfterOdd = false;

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

{

if (sequence[i] % 2 != 0 && sequence[i + 1] % 2 == 0)

{

evenAfterOdd = true;

break;

}

}

if (evenAfterOdd)

{

Console.WriteLine("Положительные числа в обратном порядке:");

for (int i = n - 1; i >= 0; i--)

{

if (sequence[i] > 0)

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

}

}

else

{

Console.WriteLine("Отрицательные числа в обратном порядке:");

for (int i = n - 1; i >= 0; i--)

{

if (sequence[i] < 0)

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

}

}

Console.ReadLine();

}

}

Таблица 1.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 5, 1, 33, 2, 77, -9 | 77, 2, 33, 1 |

Анализ результатов:

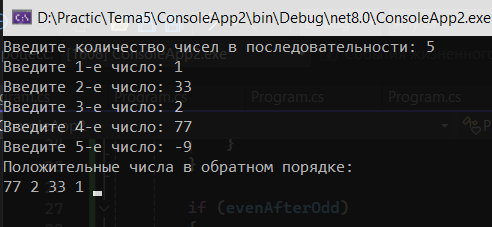


Рисунок 1.2 – Результат работы программы

Задание 3. В каждой задаче задается квадратная целочисленная матрица NxN

(значение N вводится с клавиатуры, N&lt;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.

Листинг программы:

using System;

class Program

{

static void Main()

{

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

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

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

Console.Write("Введите значение a: ");

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

Console.Write("Введите значение 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 + 1);

}

}

Console.WriteLine("Исходная матрица:");

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 значение M: ");

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

double sum = 0;

int count = 0;

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

{

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

{

if (matrix[i, j] < M)

{

sum += matrix[i, j];

count++;

}

}

}

double average = count > 0 ? sum / count : 0;

Console.WriteLine($"Среднее арифметическое чисел, меньших {M}: {average}");

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

{

int columnSum = 0;

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

{

if (matrix[i, j] > 0)

{

columnSum += matrix[i, j];

}

}

Console.WriteLine($"Сумма положительных элементов столбца {j + 1}: {columnSum}");

}

Console.ReadLine();

}

}

Таблица 1.3 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 3, 1, 2, 5 | Введите размерность матрицы (NxN): 3  Введите значение a: 1 Введите значение b: 2 Исходная матрица:  1 1 2  2 2 2  1 2 1  Введите значение M: 5 |

Продолжение таблицы 1.3

|  |  |
| --- | --- |
|  | Среднее арифметическое чисел, меньших 5: 1,5555555555555556  Сумма положительных элементов столбца 1: 4  Сумма положительных элементов столбца 2: 5  Сумма положительных элементов столбца 3: 5 |

Анализ результатов:

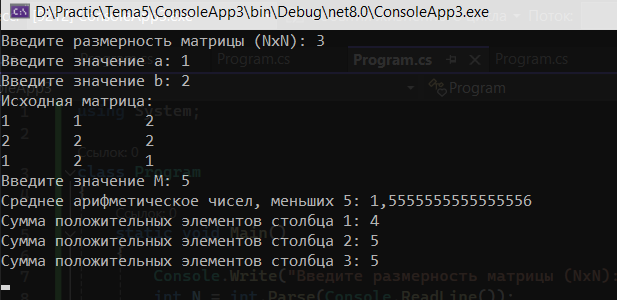


Рисунок 1.3 – Результат работы программы

Задание 4. Информация о количестве жильцов в каждой из четырех квартир каждого этажа 12-этажного дома хранится в двумерном массиве (в первой строке – информация о квартирах первого этажа, во второй – второго и т.д.). В каждой квартире проживает одна семья. Найти численность самой большой семьи в квартирах 3-го и 4-го этажей.

Листинг программы:

using System;

class Program

{

static void Main()

{

int[,] residents = new int[12, 4];

Random random = new Random();

for (int i = 0; i < 12; i++)

{

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

{

residents[i, j] = random.Next(1, 6);

}

}

int sum3rdFloor = residents[2, 0] + residents[2, 1] + residents[2, 2] + residents[2, 3];

int sum4thFloor = residents[3, 0] + residents[3, 1] + residents[3, 2] + residents[3, 3];

if (sum3rdFloor > sum4thFloor)

{

Console.WriteLine($"На 3-м этаже больше жильцов ({sum3rdFloor} человек).");

}

else if (sum4thFloor > sum3rdFloor)

{

Console.WriteLine($"На 4-м этаже больше жильцов ({sum4thFloor} человек).");

}

else

{

Console.WriteLine("На 3-м и 4-м этажах одинаковое количество жильцов.");

}

Console.ReadLine();

}

}

Таблица 1.4 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | На 4-м этаже больше жильцов (14 человек). |

Анализ результатов:

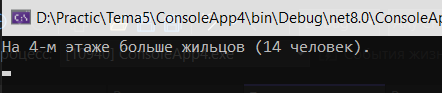


Рисунок 1.4 – Результат работы программы

Задание 5. Разработать рекурсивный алгоритм и программу решения задачи, в которой вычислить: f(n)=(1+n!)/(2+n!). Исходные данные вводятся с клавиатуры.

Листинг программы:

using System;

class Program

{

static double CalculateF(int n)

{

if (n == 0)

{

return 1;

}

else

{

double factorialN = CalculateFactorial(n);

return (1 + factorialN) / (2 + factorialN);

}

}

static double CalculateFactorial(int num)

{

if (num == 0)

{

return 1;

}

else

{

return num \* CalculateFactorial(num - 1);

}

}

static void Main()

{

Console.Write("Введите значение n: ");

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

double result = CalculateF(n);

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

Console.ReadLine();

}

}

Таблица 1.5 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 5 | f(5) = 0,9918032786885246 |

Анализ результатов:

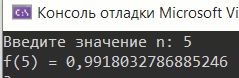


Рисунок 1.5 – Результат работы программы

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

Листинг программы:

using System;

class Program

{

static void Main()

{

int n = 7;

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

int num = 1;

int rowStart = 0;

int rowEnd = n - 1;

int colStart = 0;

int colEnd = n - 1;

while (num <= n \* n)

{

for (int i = colStart; i <= colEnd; i++)

{

spiralMatrix[rowStart, i] = num++;

}

rowStart++;

for (int i = rowStart; i <= rowEnd; i++)

{

spiralMatrix[i, colEnd] = num++;

}

colEnd--;

for (int i = colEnd; i >= colStart; i--)

{

spiralMatrix[rowEnd, i] = num++;

}

rowEnd--;

for (int i = rowEnd; i >= rowStart; i--)

{

spiralMatrix[i, colStart] = num++;

}

colStart++;

}

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

{

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

{

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

}

Console.WriteLine();

}

Console.ReadLine();

}

}

Таблица 1.6 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | 1 2 3 4 5 6 7  24 25 26 27 28 29 8  23 40 41 42 43 30 9  22 39 48 49 44 31 10  21 38 47 46 45 32 11  20 37 36 35 34 33 12  19 18 17 16 15 14 13 |

Анализ результатов:

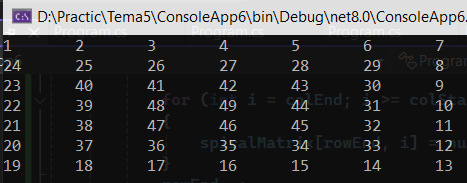


Рисунок 1.6 – Результат работы программы