**5 МАССИВЫ. АЛГОРИТМЫ ОБРАБОТКИ МАССИВОВ. РЕКУРСИЯ**

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

Листининг программы:

using System;

namespace task1

{

class Program

{

static void Main(string[] args)

{

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

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

int[] numbers = new int[size];

Random rand = new Random();

Console.WriteLine("Старый массив");

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

{

numbers[i] = rand.Next(30);

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

}

Console.WriteLine("\nНовый массив");

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

{

if (numbers[i] % 2 == 0)

{

numbers[i] = 0;

}

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

}

}

}

}

Таблица 5.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 5 | [8, 25, 13, 8, 22], [0, 25, 13, 0, 0] |

Анализ результатов:

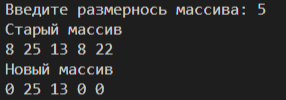


Рисунок 5.1 – Результат работы программы

Источник – собственная разработка

Задание 2. Пусть даны целые числа , ...,. 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.

Листининг программы:

using System;

namespace task\_2

{

class Program

{

static void Main(string[] args)

{

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

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

int[] numbers = new int[size];

Random rand = new Random();

bool truth = false;

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

{

numbers[i] = rand.Next(-4,5);

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

}

Console.WriteLine("");

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

{

if ((numbers[i]%2==0) && (numbers[i-1] % 2 != 0))

{

truth = true;

}

else if ((numbers[i] % 2 == 0) && (numbers[i - 1] % 2 == 0))

{

truth = false;

break;

}

}

if (truth==true)

{

for (int i = numbers.Length-1; i >= 0; i--)

{

if (numbers[i] >= 0)

{

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

}

}

}

else

{

for (int i = numbers.Length-1; i >= 0; i--)

{

if (numbers[i] < 0)

{

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

}

}

}

}

}

}

Таблица 5.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 8 | [3, -2, -4, 2, -3, 3, 2, 3], [-3, -4, -2] |

Анализ результатов:



Рисунок 5.2 – Результат работы программы

Источник – собственная разработка

Задание 3. Вычислить среднее арифметическое чисел, небольших заданного M. Вычислить сумму положительных элементов каждого столбца.

Листининг программы:

using System;

namespace task3

{

class Program

{

static void Main(string[] args)

{

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

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

int[,] numbers = new int[size,size];

Random rand = new Random();

Console.Write("Введите число начала диапазона чисел: ");

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

Console.Write("Введите число конца диапазона чисел: ");

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

Console.Write("Введите М: ");

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

double counter = 0;

double average = 0;

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

{

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

{

numbers[i, j] = rand.Next(a,b+1);

if (numbers[i, j] <= m)

{

counter++;

average += numbers[i, j];

}

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

}

Console.WriteLine("");

}

double total = average / counter;

Console.WriteLine($"среднее арифметическое чисел, небольших заданного M: {total}");

for (int i = 0; i < numbers.GetLength(1); i++)

{

int sum = 0;

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

{

if (numbers[j, i]%2==0)

{

sum += numbers[j, i];

}

}

Console.WriteLine($"сумма положительных чисел столбца {i+1}: {sum}"); ;

}

}

}

}

Таблица 5.3 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 3, 1, 3, 5 | 3 1 1  2 2 3  3 1 3, 2,111, 2, 2, 0 |

Анализ результатов:

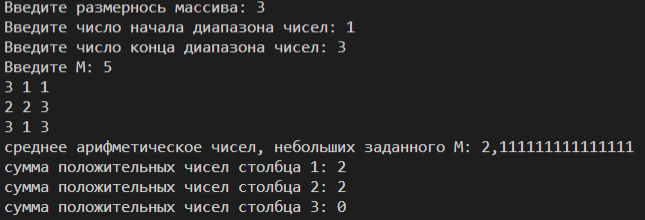


Рисунок 5.3 – Результат работы программы

Источник – собственная разработка

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.

Листининг программы:

using System;

namespace task4

{

class Program

{

static void Main(string[] args)

{

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

Random rand = new Random();

int max3 = 0;

int max4 = 0;

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

{

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

{

numbers[i, j] = rand.Next(1,6);

Console.Write($"{i+1} {j+1} : {numbers[i, j]} ");

}

Console.WriteLine("");

}

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

{

if (numbers[2, j] > max3)

{

max3 = numbers[2, j];

}

if (numbers[3, j] > max4)

{

max4 = numbers[3, j];

}

}

Console.WriteLine($"самая большая семья 3 этажа: {max3} человек");

Console.WriteLine($"самая большая семья 4 этажа: {max4} человек");

}

}

}

Таблица 5.4 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | 4 человек, 3 человек |

Анализ результатов:

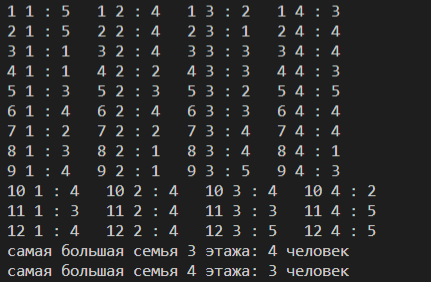


Рисунок 5.4 – Результат работы программы

Источник – собственная разработка

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

Листининг программы:

using System;

namespace task5

{

class Program

{

static double Factorial(double n)

{

if (n == 1) return 1;

return n \* Factorial(n - 1);

}

static void Main(string[] args)

{

Console.Write("Введите n: ");

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

double f = (1 + Factorial(n)) / (2 + Factorial(n));

Console.WriteLine(f);

}

}

}

Таблица 5.5 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 4 | 0,9615 |

Анализ результатов:



Рисунок 5.5 – Результат работы программы

Источник – собственная разработка