**ПРАКТИЧЕСКОЕ ЗАДАНИЕ 5**

**МАССИВЫ. АЛГОРИТМЫ ОБРАБОТКИ МАССИВОВ.**

**РЕКУРСИЯ**

Задание 1.

Найти среднее арифметическое элементов массива целых чисел

кратных пяти.

Листинг программы

using System;

namespace Array\_from\_set\_user\_parameter5

{

class Program

{

static void Main(string[] args)

{

int n;

Console.WriteLine("Введите число элементов массива");

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

double[] Data = new double[n];

int i = 0;

while (i < n)

{

Console.WriteLine("Введите элемент массива");

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

Console.WriteLine();

i++;

}

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

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

int v = 0;

int sum = 0;

Console.WriteLine("\nКратные пяти:");

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

if (Convert.ToInt32(Data[i]) % 5 == 0)

{

Console.WriteLine("Data[" + i + "]=" + Data[i]);

sum += (int)Data[i];

v += 1;

}

Console.WriteLine("Сред ариф:" + sum / v);

Console.ReadKey();

}

}

}

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 4, 45, 5, 10, 15 | Кратные пяти:  Data[0]=45  Data[1]=5  Data[2]=10  Data[3]=15  Сред ариф:18 |

Анализ результатов:

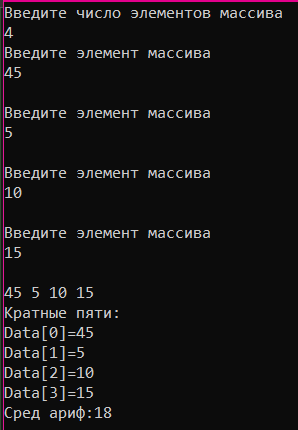


Рисунок 1.1 – Результат работы программы

Задание 2.

Листинг программы:

Console.WriteLine("Введите числа через пробел:");

int[] chisla = Console.ReadLine().Split(' ').Select(int.Parse).ToArray();

Console.WriteLine(chisla.Where((x, index) => x % 2 != 0 && index % 2 != 0).Count());

Console.ReadKey();

Таблица 1.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 4 5 6 7 | 2 |

Анализ результатов:

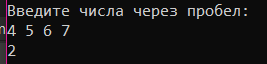


Рисунок 1.2 – Результат работы программы

Задание 3.

Вычислить сумму квадратов положительных чисел. Вычислить сумму элементов каждой строки.

Листинг программы:

try

{

Console.Write($"Введите N = ");

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

Console.Write($"Введите a = ");

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

Console.Write($"Введите b = ");

var b = int.Parse(Console.ReadLine());

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

int summI = 0;

double summPow = 0;

Random r = new Random();

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

{

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

{

matrix[i, j] = r.Next(a, b);

}

}

Console.WriteLine("\nМатрица:");

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

{

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

{

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

}

Console.WriteLine();

}

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

{

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

{

if (matrix[i, j] > 0)

{

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

}

}

Console.WriteLine($"Сумма квадратов положительных элементов строки {i} = {summPow}");

summPow = 0;

}

Console.WriteLine("\nСумма каждой строки:");

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

{

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

{

summI += matrix[i, j];

}

Console.WriteLine($"Summ {i} строки = {summI}");

summI = 0;

}

}

catch (FormatException ex)

{

Console.WriteLine(ex.Message);

}

catch (OverflowException ex)

{

Console.WriteLine(ex.Message);

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

Таблица 1.3 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Введите N = 4  Введите a = 1  Введите b = 5  Матрица:  3 3 3 4  4 1 4 2  3 1 4 3  4 3 1 4 | Сумма квадратов положительных элементов строки 0 = 43  Сумма квадратов положительных элементов строки 1 = 37  Сумма квадратов положительных элементов строки 2 = 35  Сумма квадратов положительных элементов строки 3 = 42  Сумма каждой строки:  Summ 0 строки = 13  Summ 1 строки = 11  Summ 2 строки = 11  Summ 3 строки = 12 |

Анализ результатов:

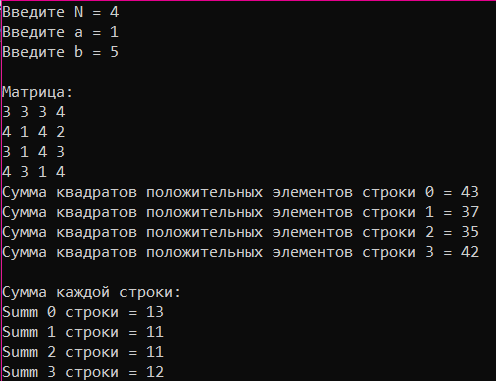


Рисунок 1.3 – Результат работы программы

Задание 4.

В поезде 20 вагонов, в каждом по 36 мест. Информация о проданных на поезд билетах хранится в двумерном массиве, номера строк которого соответствуют номерам вагонов, а номера столбцов – номерам мест. Если билет на то или иное место продан, то соответствующий элемент массива имеет значение 1, в противном случае – 0. Определить имеются ли в поезде свободные места.

Листинг программы:

namespace ConsoleApp2

{

class Program

{

static void Main(string[] args)

{

int N = 20;

int M = 36;

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

Random random = new Random();

Console.ForegroundColor = ConsoleColor.Green;

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

Console.WriteLine("Матрица из вагонов:");

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

{

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

{

int rand = random.Next(0, 2);

Matrix[i, j] = rand;

}

}

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

{

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

{

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

}

Console.WriteLine();

}

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

Console.ForegroundColor = ConsoleColor.Blue;

Console.Write("Введите номер вагона: ");

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

int x = 0;

for (int j = 0; j < Matrix.GetLength(1); j++) if (Matrix[vagon - 1, j] == 0) x++;

Console.WriteLine($" Вагон - {vagon}."

+ $" - Свободных мест в вагоне - {x}.");

Console.ReadKey();

}

}

}

Таблица 1.4 – Входные и выходные данные

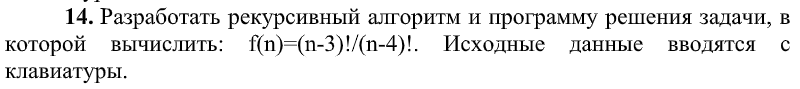
|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 18 | 19 |

Анализ результатов:



Рисунок 1.4 – Результат работы программы

Задание 5.

Листинг программы:

static int Factorial(int f)

{

if (f == 0)

return 1;

else

return f \* Factorial(f - 1);

}

try

{

Console.Write("Введите n = ");

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

var f = Factorial(n - 3) / Factorial(n - 4);

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

}

catch (FormatException e)

{

Console.WriteLine(e.Message);

}

catch (DivideByZeroException e)

{

Console.WriteLine(e.Message);

}

catch (Exception e)

{

Console.WriteLine(e.Message);

}

Таблица 1.5 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 4 | 1 |

Анализ результатов:



Рисунок 1.5 – Результат работы программы

Дополнительное задание



Листинг программы:

namespace add

{

class Program

{

static void Main(string[] args)

{

const int x = 7;

const int z = 7;

int[,] array = new int[x, z];

int row = 0;

int column = 0;

int diagonalX = 1;

int diagonalY = 0;

int Changes = 0;

int visit = z;

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

{

array[row, column] = i + 1;

if (--visit == 0)

{

visit = z \* (Changes % 2) + x \* ((Changes + 1) % 2) - (Changes / 2 - 1) - 2;

int temp = diagonalX;

diagonalX = -diagonalY;

diagonalY = temp;

Changes++;

}

column += diagonalX;

row += diagonalY;

}

Console.WriteLine("Spiral two dimensional array:");

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

{

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

{

Console.Write($"{array[i, j],3}");

}

Console.WriteLine();

}

}

}

}

Анализ результатов:

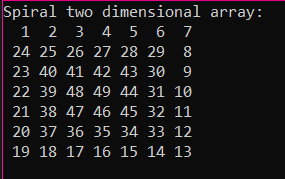


Рисунок 1.6 – Результат работы программы