**5 МАССИВЫ. КЛАСС ARRAY**

Задание №1. В массиве вещественных чисел найти минимальный элемент среди отрицательных элементов.

Листинг программы:

static double negativeChoose(double[] array)

{

return array.Where(x => x < 0).Min();

}

static void Main(String[] args)

{

Random random = new Random();

double[] array = new double[20];

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

{

array[i] = random.Next(-10, 10);

Console.WriteLine($"{i} : {array[i]}");

}

double numberOfNegative = negativeChoose(array);

Console.WriteLine($"Minimal negative element - {numberOfNegative}"); }

Таблица 5.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Array | -10 |

Источник: собственная разработка

Анализ результатов представлен на рисунке 5.1.

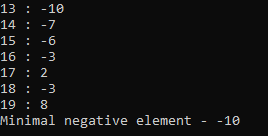


Рисунок 5.1 – Результат работы программы

Источник: собственная разработка

Задание №2 Cформировать и вывести на экран одномерный массив согласно варианту. Произвести его сортировку и бинарный поиск числа k (k - вводится с клавиатуры).

Пусть даны натуральные числа n, а1, ..., аn. Определите количество членов аk последовательности а1, ..., аn кратных 3 и не кратных 5.

Листинг программы:

public static object Find(int[] arr)

{

return arr.Where(x => x % 3 == 0 && x % 5 != 0).Count();

}

static void Main(String[] args)

{

int fromValue = 0, toValue = 0;

Console.Write("Enter first array value (from): ");

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

Console.Write("Enter second array value (to): ");

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

int[] array = new int[toValue+1];

for (fromValue = fromValue; fromValue <= toValue; fromValue++)

{

array[fromValue] = fromValue;

Console.WriteLine(array[fromValue]);

}

Console.WriteLine("Amount of numbers that are multiples of 3 and not multiples on 5: ");

Console.WriteLine(Find(array));

}

Таблица 5.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 1, 5 | 1 |

Источник: собственная разработка

Анализ результатов представлен на рисунке 5.2.

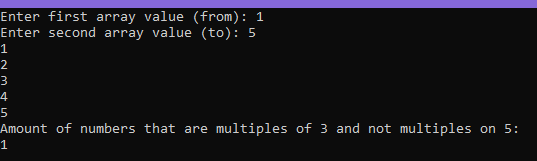


Рисунок 5.2 – Результат работы программы

Источник: собственная разработка

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.

Вычислить среднее арифметическое отрицательных чисел. Вычислить сумму элементов каждой строки.

Листинг программы:

static void Main (string[] args)

{

Console.Write("Enter N value (dimensions): ");

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

int[,] array = new int[N, N]; int a, b;

Console.Write("Enter a value (renge start): ");

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

Console.Write("Enter b value (renge end): ");

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

Random random = new Random();

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

{

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

{

array[i, j] = random.Next(a, b);

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

}

Console.WriteLine();

}

try {

Console.WriteLine($" = {Avarage(array, 5)}");

}

catch (DivideByZeroException e)

{

Console.WriteLine($"Exception - Divide By Zero! {e}");

}

Console.Write("Row sum: \n");

WorkWithMatrix(array);

}

static void WorkWithMatrix(int[,] array)

{

int[] s = new int[array.GetLength(0)];

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

{

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

s[i] += array[i, j];

Console.WriteLine($"{i} | {s[i]}");

}

}

static double Avarage(int[,] array, int G)

{

int sr = 0;

int indexer = 0;

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

{

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

{

if (array[i, j] < 0)

{

sr += array[i, j];

indexer++;

}

}

}

return sr / indexer;

}

Таблица 5.3 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 5, -5, 5 | -3, 1, 0, -14, -2, -9 |

Источник: собственная разработка

Анализ результатов представлен на рисунке 5.3.

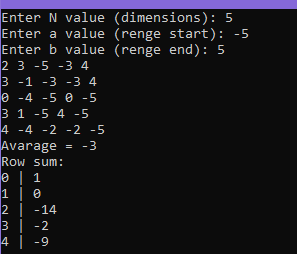


Рисунок 5.3 – Результат работы программы

Источник: собственная разработка

Задание №4. Решить задачу на проверку условий после выполнения расчетов. 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:

static void Main(string[] args)

{

Console.Write("Enter n value: ");

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

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

int a, b;

Console.Write("Enter a value: ");

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

Console.Write("Enter b value: ");

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

Random random = new Random();

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

{

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

{

array[i, j] = random.Next(a, b);

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

}

Console.WriteLine();

}

printEmptyPlaces(array);

}

static void printEmptyPlaces(int[,] array)

{

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

{

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

{

if (array[i, j] == 0)

{

Console.WriteLine($"Empty place: ({i} wagon, {j} place)"); }

}

}

}

Таблица 5.4 – Входные и выходные данные

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

Источник: собственная разработка

Анализ результатов представлен на рисунке 5.4.

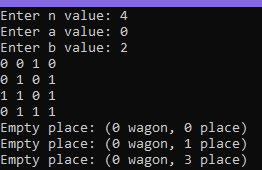


Рисунок 5.4 – Результат работы программы

Источник: собственная разработка

Задание №5. Разработать рекурсивный алгоритм и программу решения задачи, в которой вычислить: f(n)=(n-2)!/n!. Исходные данные вводятся с клавиатуры.

Листинг программы:

static void Main(string[] args)

{

Console.Write("Enter n value: ");

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

Console.WriteLine($"Expression result: {Expression(n)}");

}

static double Expression(int n)

{

return Fact(n + 2) / Fact(n);

}

static int Fact(int f)

{

if (f == 0)

return 1;

else

return f \* Fact(f - 1);

}

Таблица 5.5 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 5 | 42 |

Источник: собственная разработка

Анализ результатов представлен на рисунке 5.5.



Рисунок 5.5 – Результат работы программы

Источник: собственная разработка

Задание №6. Заполнить двумерный массив размером 7х7 числами 1, 2, …, 49, расположенными в нем по спирали.

Листинг программы:

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(); }

}

Таблица 5.6 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 7, 7 | Array |

Источник: собственная разработка

Анализ результатов представлен на рисунке 5.6.

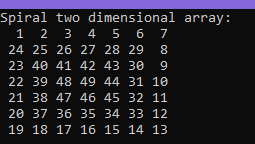


Рисунок 5.6 – Результат работы программы

Источник: собственная разработка