1. **Массивы. Алгоритмы обработки массивов. Рекурсия**

Задание 1. Вывести на экран дисплея номера элементов массива, удовлетворяющих условию.

Листинг программы:

try

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

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

double[] arr = new double[leng];

Random rnd = new Random();

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

{

arr[i] = Math.Round(rnd.NextDouble() \* 5, 2);

Console.WriteLine(arr[i]);

}

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

{

if (arr[i] > 0 && arr[i] < 3.2)

{

Console.WriteLine($"Условию 0<x<3,2 удовлетворяет: {i} элемент массива");

}

}

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

Анализ результатов:

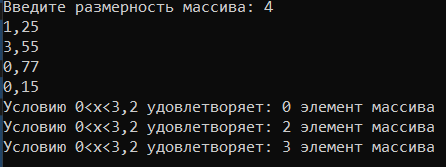


Рисунок 5.1 – Результат работы программы

Источник: собственная разработка

Задание 2. Пусть даны вещественные числа а1, ..., а20. Преобразуйте эту

последовательность по правилу: большее из аi и а10 + i (i = 1, ..., 10) примите в качестве нового значения аi, меньшее - в качестве нового значения а10 + i.

Листинг программы:

try

{

int leng = 20;

double[] arr = new double[leng];

Random rnd = new Random();

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

{

arr[i] = Math.Round(rnd.NextDouble() \* 5, 2);

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

if (i == 9)

Console.WriteLine();

}

Console.WriteLine();

for (int i = 1; i < 10; i++)

{

if (arr[i] < arr[i+10])

{

double temp = arr[i];

arr[i] = arr[i+10];

arr[i+10] = temp;

}

}

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

{

Console.Write(Math.Round(arr[i], 2) + " ");

if (i == 9)

Console.WriteLine();

}

Array.Sort(arr);

Console.WriteLine();

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

{

Console.Write(Math.Round(arr[i], 2) + " ");

}

Console.WriteLine();

Console.WriteLine("Введите число: ");

double k = Convert.ToDouble(Console.ReadLine());

int index = Array.BinarySearch(arr, k);

if (index < 0)

{

Console.WriteLine("Число не найдено");

}

else

{

Console.WriteLine($"Число {k} находится в массиве под индексом {index}");

}

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

Анализ результатов:

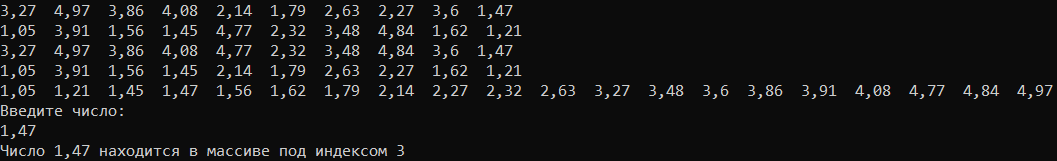


Рисунок 5.2 – Результат работы программы

Источник: собственная разработка

Задание 3. Вычислить сумму квадратов отрицательных чисел. Определить

наименьший элемент в каждой строки.

Листинг программы:

try

{

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

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

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

Console.WriteLine("Введите левую границу диапазона допустимых значений: ");

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

Console.WriteLine("Введите правую границу диапазона допустимых значений: ");

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

Random rnd = new Random();

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

{

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

{

arr[i,j] = rnd.Next(start, end + 1);

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

}

Console.WriteLine();

}

int[] ar = arr.Cast<int>().ToArray();

int summ = ar.Where(x => x < 0).Select(x => x \* x).Sum();

Console.WriteLine($"Сумма квадратов отрицаельных чисел матрицы: {summ}");

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

{

int minElem = arr[0, 0];

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

{

minElem = arr[i,j] < minElem ? arr[i,j] : minElem;

}

Console.WriteLine($"Минимальный элемент {i} строки: {minElem}");

}

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

Таблица 5.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 4, -3, 10 | -3, 1, -2, -1 |

Анализ результатов:

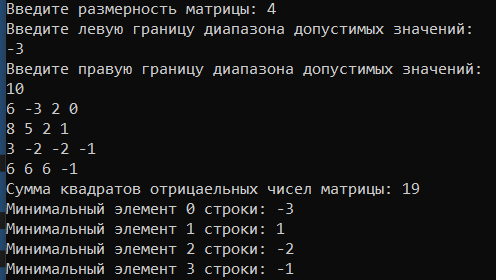


Рисунок 5.3 – Результат работы программы

Источник: собственная разработка

Задание 4. Фирма имеет 10 магазинов. Информация о доходе каждого

магазина за каждый месяц года хранится в двумерном массиве (в первом столбце за январь, во втором – за февраль и т.д.). Верно ли, что общий доход фирмы в сентябре превысил некоторое заданное число?

Листинг программы:

try

{

int row = 12;

int col = 20;

int[,] arr = new int[row, col];

Random rnd = new Random();

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

{

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

{

arr[i, j] = rnd.Next(50, 101);

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

}

Console.WriteLine();

}

int sumFeb = 0;

int sumOct = 0;

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

{

sumFeb += arr[1, i];

sumOct += arr[9, i];

}

Console.WriteLine($"Сумма зарплат в феврале: {sumFeb}");

Console.WriteLine($"Сумма зарплат в октябре: {sumOct}");

if (sumFeb < sumOct)

Console.WriteLine("Верно");

else

Console.WriteLine("Неверно");

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

Таблица 5.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| - | 1480, 1487 |

Анализ результатов:

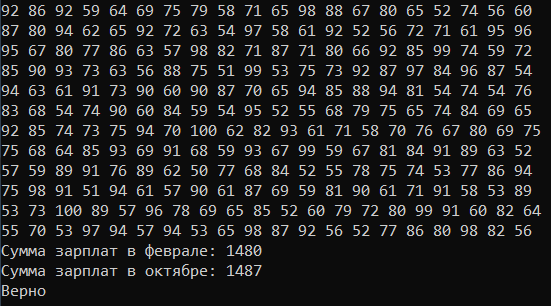


Рисунок 5.4 – Результат работы программы

Источник: собственная разработка

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

Листинг программы:

try

{

Console.Write("Введите число: ");

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

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

Console.WriteLine($"Результат выражения: {Math.Round(result, 2)}");

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

double Fact(int n)

{

if (n == 1) return 1;

return n \* Fact(n - 1);

}

Таблица 5.3 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 4 | 0,25 |

Анализ результатов:

C:\Users\1234321\Downloads\govno13.png

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:

try

{

const int n = 7;

const int m = 7;

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

int row = 0;

int col = 0;

int dx = 1;

int dy = 0;

int dirChanges = 0;

int visits = m;

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

{

matrix[row, col] = i + 1;

if (--visits == 0)

{

visits = m \* (dirChanges % 2) + n \* ((dirChanges + 1) % 2) - (dirChanges / 2 - 1) - 2;

int temp = dx;

dx = -dy;

dy = temp;

dirChanges++;

}

col += dx;

row += dy;

}

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

{

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

{

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

}

Console.WriteLine();

}

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

Анализ результатов:

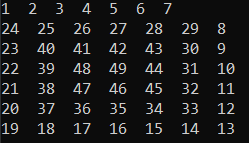


Рисунок 5.6 – Результат работы программы

Источник: собственная разработка