**ПРАКТИЧЕСКОЕ ЗАДАНИЕ 16**

**РАБОТА С ФАЙЛАМИ**

Задание №1. Компонентами файла g являются натуральные числа. Найти количество четных чисел среди компонент.

Листинг программы:

namespace job1

{

internal class Program

{

static void Main(string[] args)

{

string file = @"C:\Users\Полина\Desktop\Кпияп\Практика\practice16\job1\g.txt";

int line = 4;

var ran = new Random(0);

using (StreamWriter sw = new StreamWriter(file))

{

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

sw.WriteLine(ran.Next());

}

int count = 0;

using (StreamReader sr = new StreamReader(file))

{

while (!sr.EndOfStream)

{

int temp = int.Parse(sr.ReadLine());

if (temp % 2 == 0)

count++;

}

}

Console.WriteLine("Количество четных: " + count);

Console.ReadKey();

}

}

}

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| file.txt | 3 |

Источник: собственная разработка

Анализ результатов представлен на рисунке 1.1.



Рисунок 1.1 – Результат работы программы

Источник: собственная разработка

Задание №2. С помощью метода CreateDirectory() создать папку “New\_folder”.

Листинг программы:

namespace job2

{

internal class Program

{

static void Main(string[] args)

{

var directory = Directory.CreateDirectory(@"C:\Users\Полина\Desktop\Кпияп\Практика\practice16\job2\Folder\_1");

}

}

}

Таблица 1.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | New\_Folder |

Источник: собственная разработка

Анализ результатов представлен на рисунке 1.2.



Рисунок 1.2 – Результат работы программы

Источник: собственная разработка

Задание №3. Создать текстовый файл, в который записать 5 строк различной длины.

Для данного файла выполнить следующие действия:

вывести весь файл на экран;

подсчитать количество строк;

подсчитать количество символов в каждой строке;

удалить последнюю строку из файла, результат записать в новый файл;

вывести на экран строки с s1 по s2;

найти длину самой длинной строки и вывести ее на экран;

вывести на экран все строки начинающиеся с заданной буквы;

переписать его строки в другой файл, порядок строк во втором файле должен быть обратным по отношению к порядку строк в заданном файле.

Листинг программы:

try

{

string path = @"C:\Users\Полина\Desktop\Кпияп\Практика\practice16\job3\test.txt";

using (StreamReader sr = new StreamReader(path))

{

PrintFile(sr);

}

string[] str = StringSplit(path);

Console.WriteLine(str.Length);

int[] arr = CharCount(str);

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

Console.WriteLine($"В {i} строке {arr[i]} символов");

DeleteLast(str);

Console.Write("Введите начальную строку: ");

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

Console.Write("Введите конечную строку: ");

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

PrintFromRange(str, start, end);

StringMax(str);

Console.Write("Введите букву: ");

char letter = Convert.ToChar(Console.ReadLine());

PrintFromLetter(str, letter);

PrintInFile(str);

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

void PrintFile(StreamReader sr)

{

Console.WriteLine(sr.ReadToEnd());

}

string[] StringSplit(string path)

{

string[] str = File.ReadAllLines(path);

return str;

}

int[] CharCount(string[] str)

{

int[] arr = new int[str.Length];

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

arr[i] = str[i].Length;

return arr;

}

void DeleteLast(string[] str)

{

string[] st = new string[str.Length - 1];

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

st[i] = str[i];

File.WriteAllLines(@"C:\Users\Полина\Desktop\Кпияп\Практика\practice16\job3\test.txt", st);

}

void PrintFromRange(string[] str, int start, int end)

{

for (int i = start; i <= end; i++)

{

Console.WriteLine(str[i]);

}

}

void StringMax(string[] str)

{

Console.WriteLine(str.Select((x) => x.Length).Max());

}

void PrintFromLetter(string[] str, char letter)

{

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

if (str[i][0] == letter)

Console.WriteLine(str[i]);

}

void PrintInFile(string[] str)

{

Array.Reverse(str);

StreamWriter sw = new StreamWriter(@"C:\Users\Полина\Desktop\Кпияп\Практика\practice16\job3\test1.txt");

foreach (string st in str)

{

sw.WriteLine(st);

}

sw.Close();

}

Таблица 1.3 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| file.txt | Никита  Полина  Инна  Настя  Толочко Павел Станиславович  5  В 0 строке 6 символов  В 1 строке 6 символов  В 2 строке 4 символов  В 3 строке 5 символов  В 4 строке 27 символов  Введите начальную строку: 1  Введите конечную строку: 2  Полина  Инна  27 |

Источник: собственная разработка

Анализ результатов представлен на рисунке 1.3.

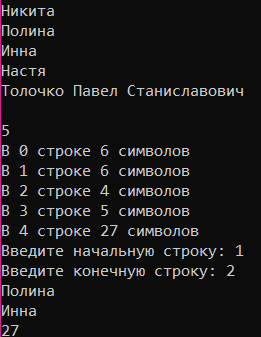


Рисунок 16.3 – Результат работы программы

Источник: собственная разработка

Задание №4. Имеются два текстовых файла с одинаковым числом строк. Выяснить совпадают ли их строки. Если нет, то получить номер первой строки, в которой эти файлы отличаются друг от друга.

Листинг программы:

using System.Text;

var firstFilePath = @"C:\Users\Полина\Desktop\Кпияп\Практика\practice16\job4\firstD.txt";

var secondFilePath = @"C:\Users\Полина\Desktop\Кпияп\Практика\practice16\job4\secondD.txt";

if (!File.Exists(firstFilePath))

{

File.Create(firstFilePath);

}

if (!File.Exists(secondFilePath))

{

File.Create(secondFilePath);

}

var linesFromFile1 = File.ReadAllLines(firstFilePath);

var linesFromFile2 = File.ReadAllLines(secondFilePath);

if (linesFromFile1.Count() != linesFromFile2.Count())

{

Console.WriteLine("Файлы разные по длине");

}

else

{

Console.WriteLine("Файл одинаковые по длине");

var index = FindFirstIndex(linesFromFile1, linesFromFile2);

if (index == -1)

{

Console.WriteLine("Файлы одинаковые");

}

else

{

Console.WriteLine($"Файлы разные в строке {index}");

}

}

int FindFirstIndex(string[] linesFromFile1, string[] linesFromFile2)

{

var index = -1;

for (int i = 0; i < linesFromFile1.Count(); i++)

{

if (!linesFromFile1[i].Equals(linesFromFile2[i]))

{

return i + 1;

}

}

return index;

}

Таблица 1.4 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| f1, f2 | Файлы разные по длине |

Источник: собственная разработка

Анализ результатов представлен на рисунке 1.4.



Рисунок 1.4 – Результат работы программы

Источник: собственная разработка

Задание №5. Даны два файла целых чисел f1.dat и f2.dat (предварительно эти файлы создайте программно). Необходимо:

найти в файле f2.dat число, наиболее близкое по величине к минимальному значению в файле f2.dat;

определить в каком из файлов больше положительных, отрицательных и нулевых значений.

определить, являются ли числа в файле f1.dat упорядоченными по возрастанию.

определить, образуют ли числа в файле f1.dat знакопеременную последовательность чисел.

Листинг программы:

class Program

{

static void Main()

{

FileStream file1 = new FileStream(@"C:\Users\Полина\Desktop\Кпияп\Практика\practice16\job5\file1.dat", FileMode.Create, FileAccess.ReadWrite);

StreamWriter writer1 = new StreamWriter(file1);

FileStream file2 = new FileStream(@"C:\Users\Полина\Desktop\Кпияп\Практика\practice16\job5\file2.dat", FileMode.Create, FileAccess.ReadWrite);

StreamWriter writer2 = new StreamWriter(file2);

Console.WriteLine("Введите количество элементов в первом файле");

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

int[] arr;

Console.WriteLine("Введите начало диапазона");

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

Console.WriteLine("Введите конец диапазона");

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

Random rnd = new Random();

arr = new int[first];

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

{

arr[i] = rnd.Next(nach, konec);

}

Array.Sort(arr);

Console.WriteLine("Первый массив");

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

{

Console.Write("{0} ", arr[i]);

writer1.Write($"{arr[i]} ");

}

writer1.Close();

arr = new int[0];

Console.WriteLine("\nВведите количество элементов во втором файле");

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

Console.WriteLine("Введите начало диапазона");

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

Console.WriteLine("Введите конец диапазона");

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

arr = new int[second];

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

{

arr[i] = rnd.Next(nachh, konecc);

}

Array.Sort(arr);

Console.WriteLine("Второй массив");

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

{

Console.Write("{0} ", arr[i]);

writer2.Write($"{arr[i]} ");

}

writer2.Close();

FileStream file4 = new FileStream(@"C:\Users\Полина\Desktop\Кпияп\Практика\practice16\job5\file1.dat", FileMode.Open, FileAccess.Read);

StreamReader reader1 = new StreamReader(file4);

string[] s1;

s1 = reader1.ReadLine().Split();

int min1 = int.Parse(s1[0]);

int min2 = int.Parse(s1[0]);

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

{

if (min1 > int.Parse(s1[i]))

{

min1 = int.Parse(s1[i]);

}

}

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

{

if (min2 > int.Parse(s1[i]) && int.Parse(s1[i]) != min1 || min2 == min1)

{

min2 = int.Parse(s1[i]);

}

}

Console.WriteLine();

Console.WriteLine("\nБлижайшее к минимальному:{0}", min2);

arr = new int[first];

Array.Resize(ref s1, s1.Length - 1);

arr = Array.ConvertAll(s1, int.Parse);

int nul1 = 0;

int polozh1 = 0;

int otric1 = 0;

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

{

if (arr[i] == 0)

nul1++;

if (arr[i] > 0)

polozh1++;

else otric1++;

}

Console.WriteLine("\nПоложительныx в первом файле: {0}", polozh1);

Console.WriteLine("\nОтрицательныx в первом файле: {0}", otric1);

Console.WriteLine("\nНулей в первом файле: {0}", nul1);

Console.WriteLine();

int t = 0;

int kol = 1;

for (int i = 0; i < arr.Length - 1; i++)

{

if (arr[i] < arr[i + 1] || arr[i] == arr[i + 1])

{

t = arr[i];

arr[i] = arr[i + 1];

arr[i + 1] = t;

kol++;

}

}

Boolean f = true;

for (int i = 0; i < arr.Length - 1; i++)

if ((arr[i] > 0 && arr[i + 1] < 0) || (arr[i] < 0 && arr[i + 1] > 0))

continue;

else

{

f = false;

break;

}

Console.WriteLine(f ? "Знакочередующаяся последовательность" : "Не знакочередующаяся последовательность");

reader1.Close();

Console.WriteLine();

if (kol == arr.Length)

Console.WriteLine("Элементы в первом файле упорядочены");

else Console.WriteLine("Элементы во втором файле не упорядочены");

FileStream file5 = new FileStream(@"C:\Users\Полина\Desktop\Кпияп\Практика\practice16\job5\file2.dat", FileMode.Open, FileAccess.Read);

StreamReader reader2 = new StreamReader(file5);

arr = new int[second];

string[] s2 = reader2.ReadLine().Split();

Array.Resize(ref s2, s2.Length - 1);

arr = Array.ConvertAll(s2, int.Parse);

int nul2 = 0;

int polozh2 = 0;

int otric2 = 0;

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

{

if (arr[i] == 0)

nul2++;

if (arr[i] > 0)

polozh2++;

else otric2++;

}

Console.WriteLine("\nПоложительныx во втором файле: {0}", polozh2);

Console.WriteLine("\nОтрицательныx во втором файле: {0}", otric2);

Console.WriteLine("\nНулей во втором файле: {0}", nul2);

reader2.Close();

Console.WriteLine();

if (nul1 > nul2)

Console.WriteLine("Нулей больше в первом файле");

if (nul1 < nul2)

Console.WriteLine("Нулей больше во втором файле");

if (nul1 == nul2)

Console.WriteLine("Нулей одинаково");

if (polozh1 > polozh2)

Console.WriteLine("Положительных больше в первом файле");

if (polozh1 < polozh2)

Console.WriteLine("Положительных больше во втором файле");

if (polozh1 == polozh2)

Console.WriteLine("Положительных одинаково");

if (otric1 > otric2)

Console.WriteLine("Отрицательных больше в первом файле");

if (otric1 < otric2)

Console.WriteLine("Отрицательных больше во втором файле");

if (otric1 == otric2)

Console.WriteLine("Отрицательных одинаково");

Console.ReadKey();

}

}

}

Таблица 16.5 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 4  1  4  5  4  10 | Введите количество элементов в первом файле  4  Введите начало диапазона  1  Введите конец диапазона  4  Первый массив  1 2 3 3  Введите количество элементов во втором файле  5  Введите начало диапазона  4  Введите конец диапазона  10  Второй массив  4 6 8 9 9  Ближайшее к минимальному:2  Положительныx в первом файле: 4  Отрицательныx в первом файле: 0  Нулей в первом файле: 0  е знакочередующаяся последовательность  Элементы в первом файле упорядочены  Положительныx во втором файле: 5  Отрицательныx во втором файле: 0  Нулей во втором файле: 0  Нулей одинаково  Положительных больше во втором файле  Отрицательных одинаково |

Источник: собственная разработка

Анализ результатов представлен на рисунке 1.5.

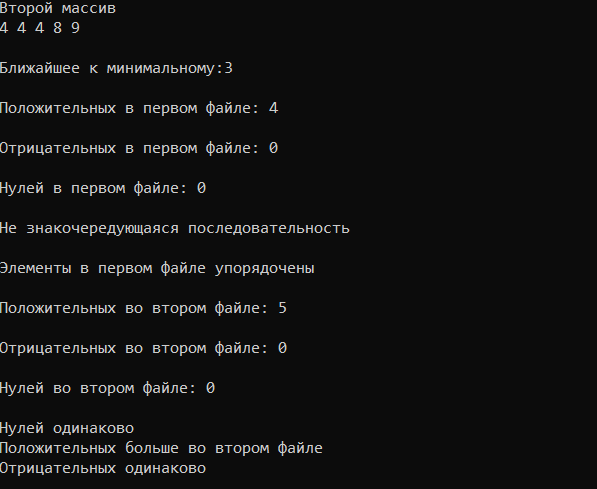


Рисунок 1.5 – Результат работы программы

Источник: собственная разработка

Задание №6. Дано предложение (вводится с клавиатуры). Слова в предложении разделяются одним или несколькими пробелами. В предложении могут встречаться знаки препинания и цифры.

Создать текстовый файл и записать в него все символы данного предложения, отличные от цифр. Вывести содержимое файла на экран.

Листинг программы:

using System.Text;

using System.Text.RegularExpressions;

string str = @"C:\Users\Полина\Desktop\Кпияп\Практика\practice16\job6\text.txt";

List<string> PaymentInf = new List<string>();

string[] Payments = File.ReadAllLines(str, Encoding.Default);

for (int x = 0; x < Payments.Length; x++)

{

Console.WriteLine($"Входная строка: {Payments[x]}");

var tttt = Payments[x].Split(new[] { ',', ' ', '-', '.' }).ToList();

char letter = 'l';

var res = Regex.Matches(Payments[x], "\\b" + letter + @"\w\*(-\w+)\*", RegexOptions.IgnoreCase).Cast<Match>().Select(i => i.Value).ToArray();

Console.WriteLine($"Слова, начинающиеся на символ {letter}:");

Console.WriteLine(string.Join("\r\n", res));

PaymentInf.Add(Payments[x]);

}

int len = 3;

Console.WriteLine($"Слова, имеющие длину {len} символа(ов):");

foreach (string R in PaymentInf)

{

string[] mas = R.Split(new Char[] { ' ', ',', '.' });

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

{

if (mas[i].Length == len)

{

Console.WriteLine(mas[i]);

}

}

}

Console.WriteLine($"Слова, начинающиеся и заканчивающиеся на одну и ту же букву:");

foreach (string R in PaymentInf)

{

string[] slova = R.Split(' ', ',', '.', '-');

foreach (string slovo in slova)

{

if (slovo.Length > 1)

{

if (slovo[0] == slovo[slovo.Length - 1])

{

Console.WriteLine(slovo);

}

}

}

}

Console.WriteLine($"Слова, начинающиеся на ту же букву, что и последнее слово:");

foreach (string R in PaymentInf)

{

string lastWord = R.Split(' ', ',', '.', '-').Last();

var matches = Regex.Matches(R, @"\b[\w]+\b", RegexOptions.Compiled);

var lastMatch = matches.Cast<Match>().Last();

string lastRemoved = R.Remove(lastMatch.Index, lastMatch.Length);

string[] slova = lastRemoved.Split(' ', ',', '.', '-');

foreach (string slovo in slova)

{

if (slovo.Length > 1)

{

if (slovo[0] == lastWord[0])

{

Console.WriteLine(slovo);

}

}

}

}

Таблица 1.6 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| file.txt | strings |

Источник: собственная разработка

Анализ результатов представлен на рисунке 1.6.

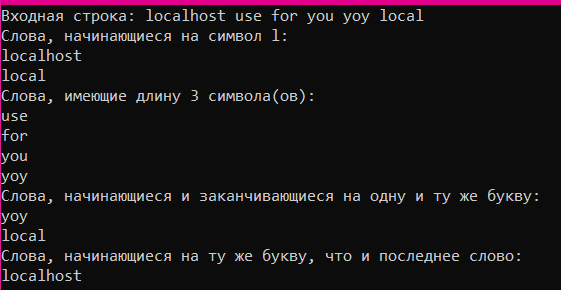


Рисунок 1.6 – Результат работы программы

Источник: собственная разработка