|  |  |  |  |
| --- | --- | --- | --- |
| **МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  **КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені Тараса Шевченка**  **ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**  **Кафедра програмних систем і технологій**  **Дисципліна**  **«Кросплатформне програмування»**  **Лабораторна робота 1** | | | |
| **Виконав:** | Кивлюк Юрій Олегович | **Перевірив**: | Петрівський В.Я |
| Група | ІПЗ-41 | Дата перевірки |  |
| Форма навчання | денна | Оцінка |  |
| Спеціальність | 121 |
| 2023 | | | |

**Варіант – 62**

Посилання на виконану роботу – [Github](https://github.com/yura-kiv/crossplatform-uni/tree/master/lab_1).

**Умова завдання:**

Марсіяни Мишко та Маша вирішили разом підібрати подарунок на день народження Каті. Коли вони нарешті знайшли те, що хотіли, та запакували предмет у гарну коробку, треба було вирішити, як підписати подарунок. Друзі подумали, що найкращим рішенням буде скласти загальний підпис так, щоб у ньому як підрядки містилися їхні імена. Майте на увазі, що на Марсі прийнято підписуватися повними іменами, а вони у марсіан можуть бути досить довгими.

Вхідні дані:

Вхідний файл INPUT.TXT містить два рядки, де записані повні імена друзів. Імена, як не дивно, складаються з літер англійського алфавіту, з яких тільки перша - велика. Довжина імен від 1 до 1000 символів.

Вихідні дані

У вихідний файл OUTPUT.TXT виведіть найкоротший рядок, у якому зустрічаються імена Миші та Маші одночасно. Літери, з яких імена починаються в цьому рядку, потрібно зробити великими. Якщо існує кілька рішень, виведіть те, яке менше в алфавітному порядку (слід вважати, що будь-яка буква у верхньому регістрі менша, ніж будь-яка буква в нижньому регістрі).

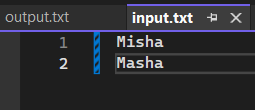
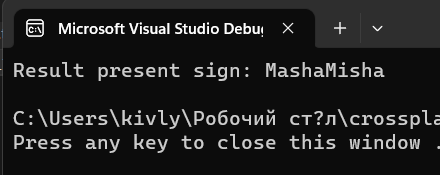
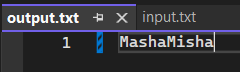
Приклади:

|  |  |  |
| --- | --- | --- |
| № | INPUT.TXT | OUTPUT.TXT |
| 1 | MishaMasha | MashaMisha |
| 2 | JulyaLyalya | JuLyalya |

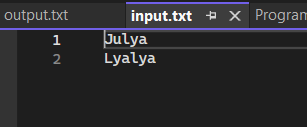
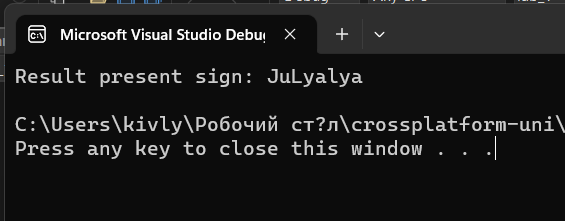
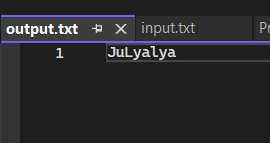
**Виконання завдання:**

Результат виконання:

1)

2)

Реалізований код:

//

// VAR - 62, Kivlyuk Yuriy IPZ-41

//

using System;

using System.IO;

using System.Text.RegularExpressions;

class Program

{

const string INPUT\_FILE = "../../../input.txt";

const string OUTPUT\_FILE = "../../../output.txt";

static bool IsValidInput(string input)

{

if (string.IsNullOrEmpty(input))

{

Console.WriteLine("You cannot enter an empty line into a program.");

return false;

}

if (!Regex.IsMatch(input, "^[a-zA-Z0-9]+$"))

{

Console.WriteLine("One or both lines contain invalid characters.");

return false;

}

return true;

}

static int CheckStrInStr(string str1, string str2)

{

int index = str1.IndexOf(str2);

if (index == -1)

return 0;

str1 = str1.Substring(0, 1).ToUpper() + str1.Substring(1);

str1 = str1.Remove(index, 1).Insert(index, str1[index].ToString().ToUpper());

Console.Write(str1);

return 1;

}

static int CalcStrInStr(string str1, string str2)

{

int str1Lenght = str1.Length;

int str2Lenght = str2.Length;

bool flag;

for (int j = 1; j < str1Lenght; j++)

{

flag = true;

for (int i = 0; (i < str1Lenght - j) && (i < str2Lenght); i++)

{

if (str1[j + i] != str2[i])

{

flag = false;

break;

}

}

if (flag)

return j;

}

return str1Lenght;

}

static string GetResult(string str1, string str2, int index)

{

str1 = str1.Substring(0, 1).ToUpper() + str1.Substring(1);

str2 = str2.Substring(0, 1).ToUpper() + str2.Substring(1);

str1 = str1.Remove(index);

Console.Write(str1);

Console.Write(str2);

return str1 + str2;

}

static void Main()

{

try

{

using (StreamReader reader = new StreamReader(INPUT\_FILE))

{

using (StreamWriter writer = new StreamWriter(OUTPUT\_FILE))

{

string name1 = reader.ReadLine();

string name2 = reader.ReadLine();

if (!IsValidInput(name1) || !IsValidInput(name2))

{

return;

}

string name1Lower = name1.ToLower();

string name2Lower = name2.ToLower();

if (CheckStrInStr(name1Lower, name2Lower) == 1 || CheckStrInStr(name2Lower, name1Lower) == 1)

return;

int i1 = CalcStrInStr(name1Lower, name2Lower);

int i2 = CalcStrInStr(name2Lower, name1Lower);

if (i1 < i2)

{

string result = GetResult(name1Lower, name2Lower, i1);

writer.Write(result);

}

else

{

string result = GetResult(name2Lower, name1Lower, i2);

writer.Write(result);

}

}

}

}

catch {

Console.WriteLine("There was a problem reading the file..");

}

}

}