### Кам’янець-Подільський національний університет імені Івана Огієнка

### КАФЕДРА КОМП’ЮТЕРНИХ НАУК

### Навчальна дисципліна «Об’єктно-орієнтоване програмування»

### ЛАБОРАТОРНА РОБОТА #0111

## Тема:

Обробка рядків і тексту. Робота з датою і часом

Варіант №2

**Виконав**:  
студент 1-го курсу  
групи KNms1-B23  
Веретко С. І.

**Прийняла**:  
старший викладач,  
Мястковська М.О.

Кам’янець-Подільський – 2023

1. Теоретичні відомості

## Обробка рядків і тексту

Рядок є об'єктом типу String, значенням якого є текст. По суті, текст зберігається у вигляді послідовної доступною тільки для читання колекції об'єктів Char. В кінці рядка в мові C# відсутній символ, який закінчується на NULL; тому рядок C# може містити будь-яке число символів NULL ( "\0"). Властивість рядка Length представляє число об'єктів Char, що містяться в цьому рядку, а не число символів Юнікоду. Для доступу до окремих точок в рядку коду Юникода використовується об'єкт StringInfo object.

## Оголошення і ініціалізація рядків

Оголошення та ініціалізацію рядків можна виконувати різними способами, як показано в наступному прикладі:

// Оголошення без ініціалізації

string message1;

// Ініціалізація в null.

string message2 = null;

// Ініціалізувати як порожній рядок.

string message3 = System.String.Empty;

// Ініціалізація рядковим значенням.

string oldPath = "c:\\Program Files\\Microsoft Visual Studio";

// Ініціалізувати як дослівний рядковий літерал.

string newPath = @"c:\Program Files\Microsoft Visual Studio";

// Використання System.String.

System.String greeting = "Hello World!";

// У локальних змінних (тобто в межах методу), можна використовувати неявні типи

var temp = "I'm still a strongly-typed System.String!";

//Використовуйте константну змінну, щоб запобігти присвоєнню іншого значення змінній.

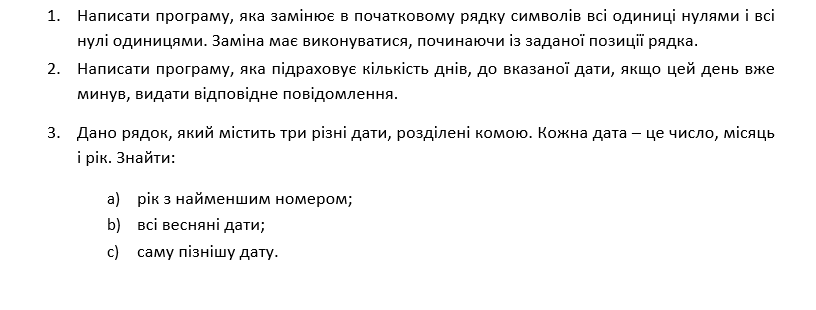
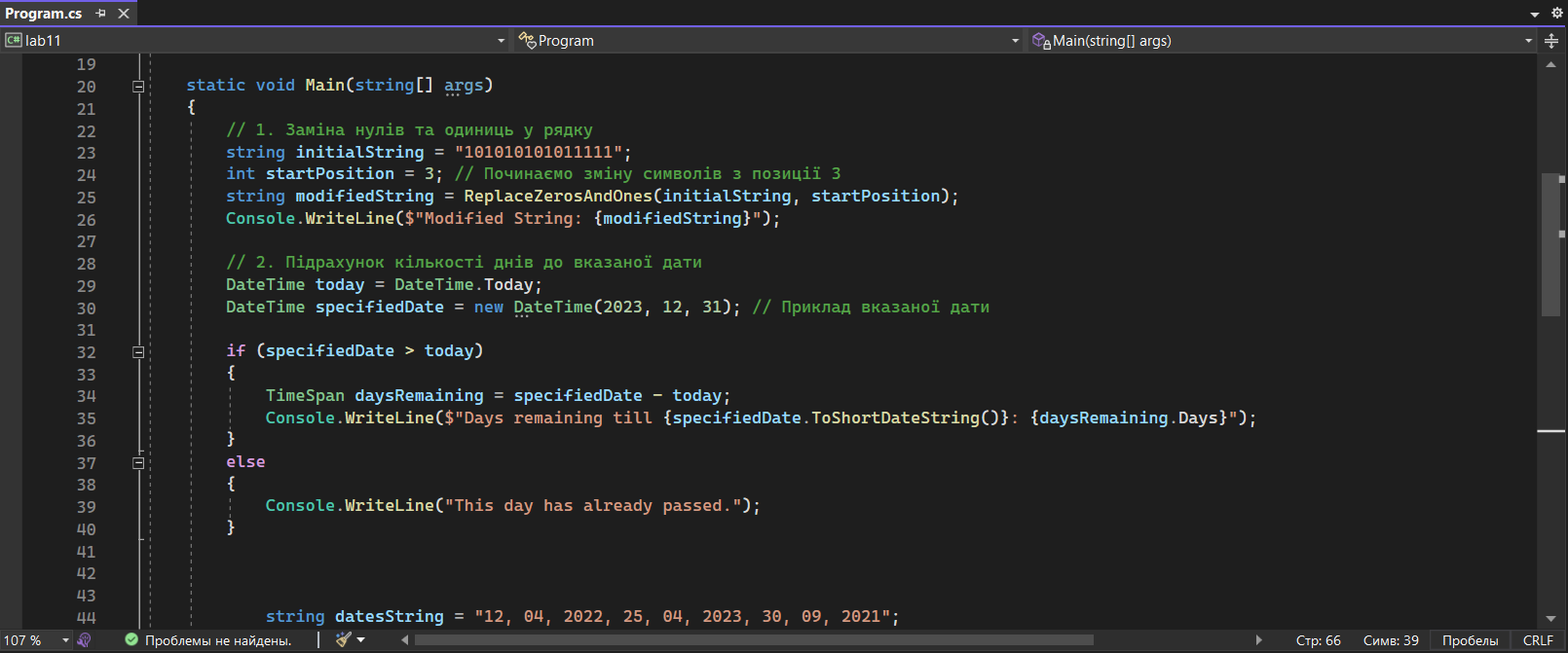
const string message4 = "You can't get rid of me!";

//Використовуйте конструктор String, тільки при створенні рядка з char\*, char[], or sbyte\*.

char[] letters = { 'A', 'B', 'C' };

string alphabet = new string(letters);

2. Завдання для виконання

  
  
3. Послідовність виконання завданнь  


4. Лістинг програми

using System;

using System.Globalization;

using System.Linq;

class Program

{

static string ReplaceZerosAndOnes(string input, int startPosition)

{

char[] chars = input.ToCharArray();

for (int i = startPosition; i < chars.Length; i++)

{

if (chars[i] == '0')

chars[i] = '1';

else if (chars[i] == '1')

chars[i] = '0';

}

return new string(chars);

}

static void Main(string[] args)

{

// 1. Заміна нулів та одиниць у рядку

string initialString = "101010101011111";

int startPosition = 3; // Починаємо зміну символів з позиції 3

string modifiedString = ReplaceZerosAndOnes(initialString, startPosition);

Console.WriteLine($"Modified String: {modifiedString}");

// 2. Підрахунок кількості днів до вказаної дати

DateTime today = DateTime.Today;

DateTime specifiedDate = new DateTime(2023, 12, 31); // Приклад вказаної дати

if (specifiedDate > today)

{

TimeSpan daysRemaining = specifiedDate - today;

Console.WriteLine($"Days remaining till {specifiedDate.ToShortDateString()}: {daysRemaining.Days}");

}

else

{

Console.WriteLine("This day has already passed.");

}

string datesString = "12, 04, 2022, 25, 04, 2023, 30, 09, 2021";

string[] dateParts = datesString.Split(", ")

.Select(part => part.Trim()).ToArray();

// Створюэмо масив дат

DateTime[] dates = new DateTime[dateParts.Length / 3]; int dateIndex = 0;

//Заповнюємо наш масив даним із масиву стрінг

for (int i = 0; i < dateParts.Length; i += 3)

{

string day = dateParts[i]; string month = dateParts[i + 1];

string year = dateParts[i + 2]; string dateString = $"{day} {month} {year}";

DateTime parsedDate = DateTime.ParseExact(dateString, "dd MM yyyy", CultureInfo.InvariantCulture); dates[dateIndex] = parsedDate;

dateIndex++;

}

//a) Знайдемо рік з найменшим номером

int minYear = dates.Select(date => date.Year).

Min(); Console.WriteLine($"Year with the smallest number: {minYear}");

// b) Знайдемо всі весняні дати

var springDates = dates.Where(date => date.Month > 2 && date.Month < 5);

Console.WriteLine("\nSpring dates : "); foreach (var springDate in springDates)

{

Console.WriteLine($"Date : {springDate.ToShortDateString()}");

}

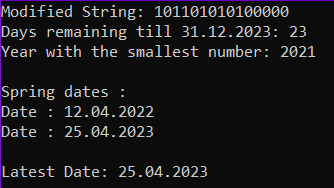
// c) Знайдемо саму пізню дату

var latestDate = dates.Max(); Console.WriteLine($"\nLatest Date: {latestDate.ToShortDateString()}");

}

}

5. Результат роботи програми



6. [xanax4rl/lab (github.com)](https://github.com/xanax4rl/lab)