

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЧЕРКАСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ БОГДАНА ХМЕЛЬНИЦЬКОГО  
Факультет обчислювальної техніки, інтелектуальних та управляючих систем  
Кафедра інформаційних технологій

**Звіт**  
**з лабораторної роботи №1**  
**Тема: Організація програми для підрахунку суми цілих чисел у**  
**бінарному вигляді.**  
**Варіант: 7**

Виконала:

студентка групи КС-242-1 Гук Марія Олегівна

Перевірив:

ст. викладач кафедри ПЗАС Мисник Б.В.

## **ЗАВДАННЯ**

Розробити програму для виконання додавання двох 7-бітних бінарних чисел з перевіркою введених даних та відображенням результату у десятковій та бінарній системах числення.

## **БЛОК-СХЕМИ**

Програма для додавання бінарних чисел складається з 5-ти методів:

- Main;
- Start (Рис.1.1);
- Check (Рис.1.2);
- SumBinary (Рис.1.3);
- Decimal (Рис.1.4).

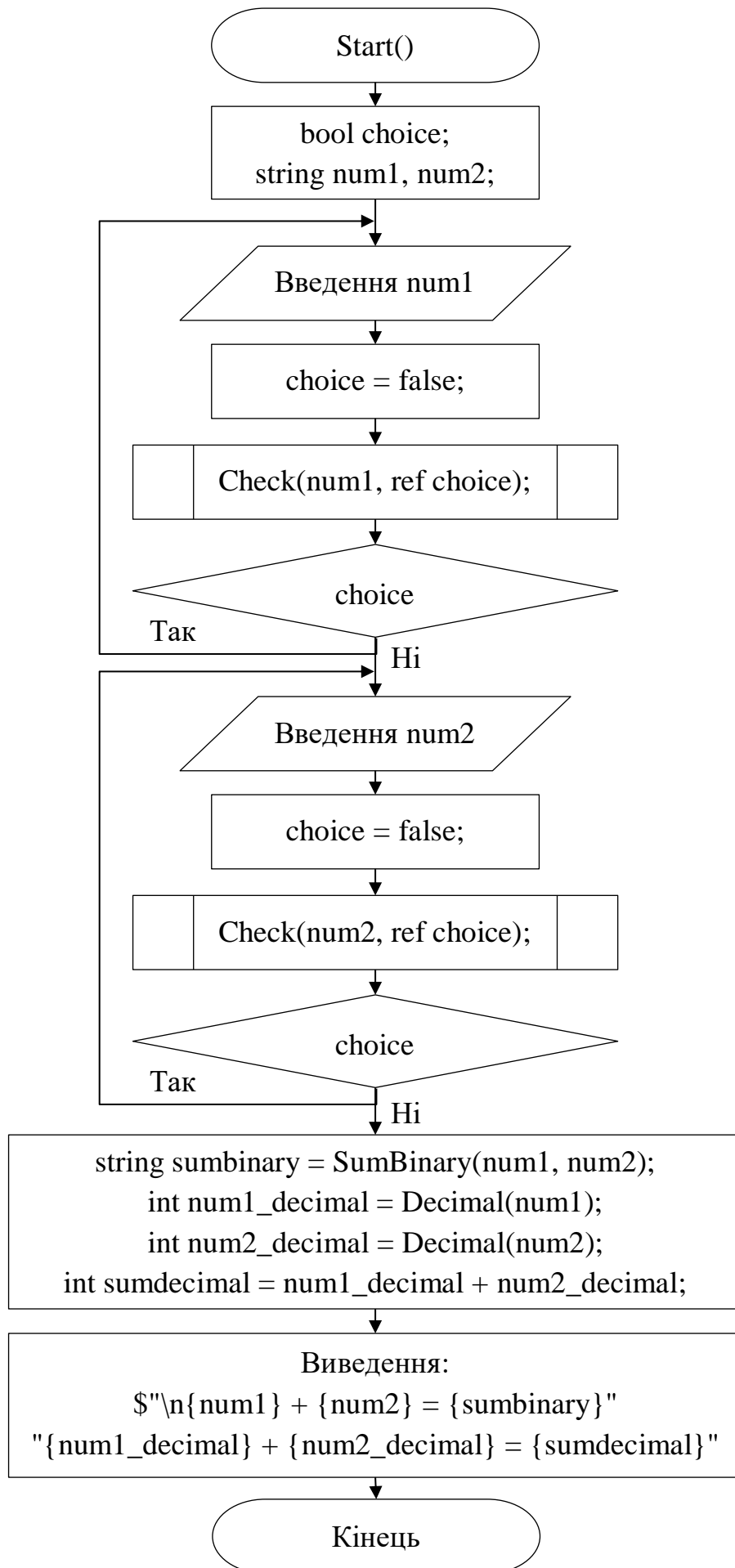


Рис. 1.1. Метод Start()

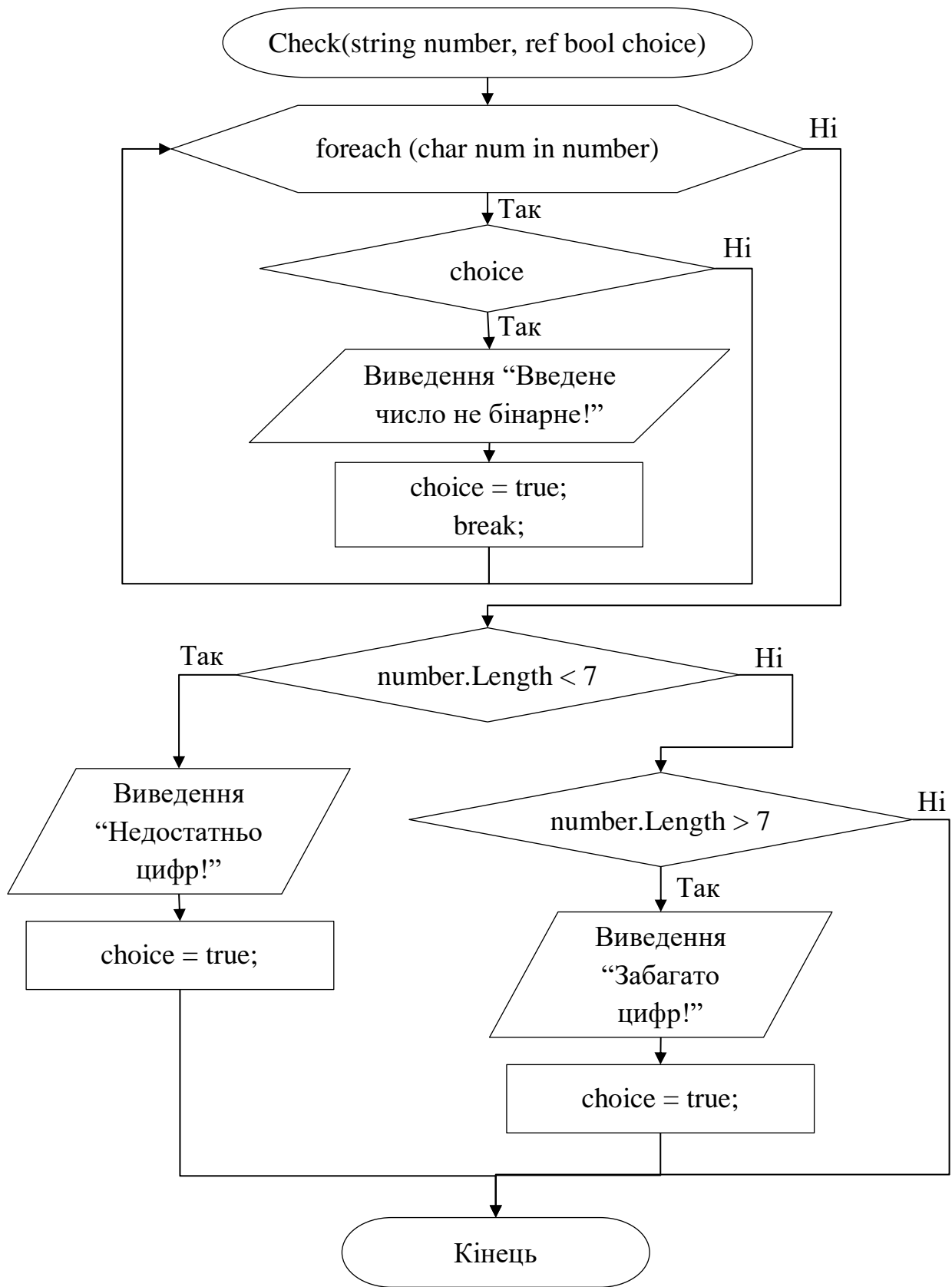


Рис. 1.2. Метод Check()

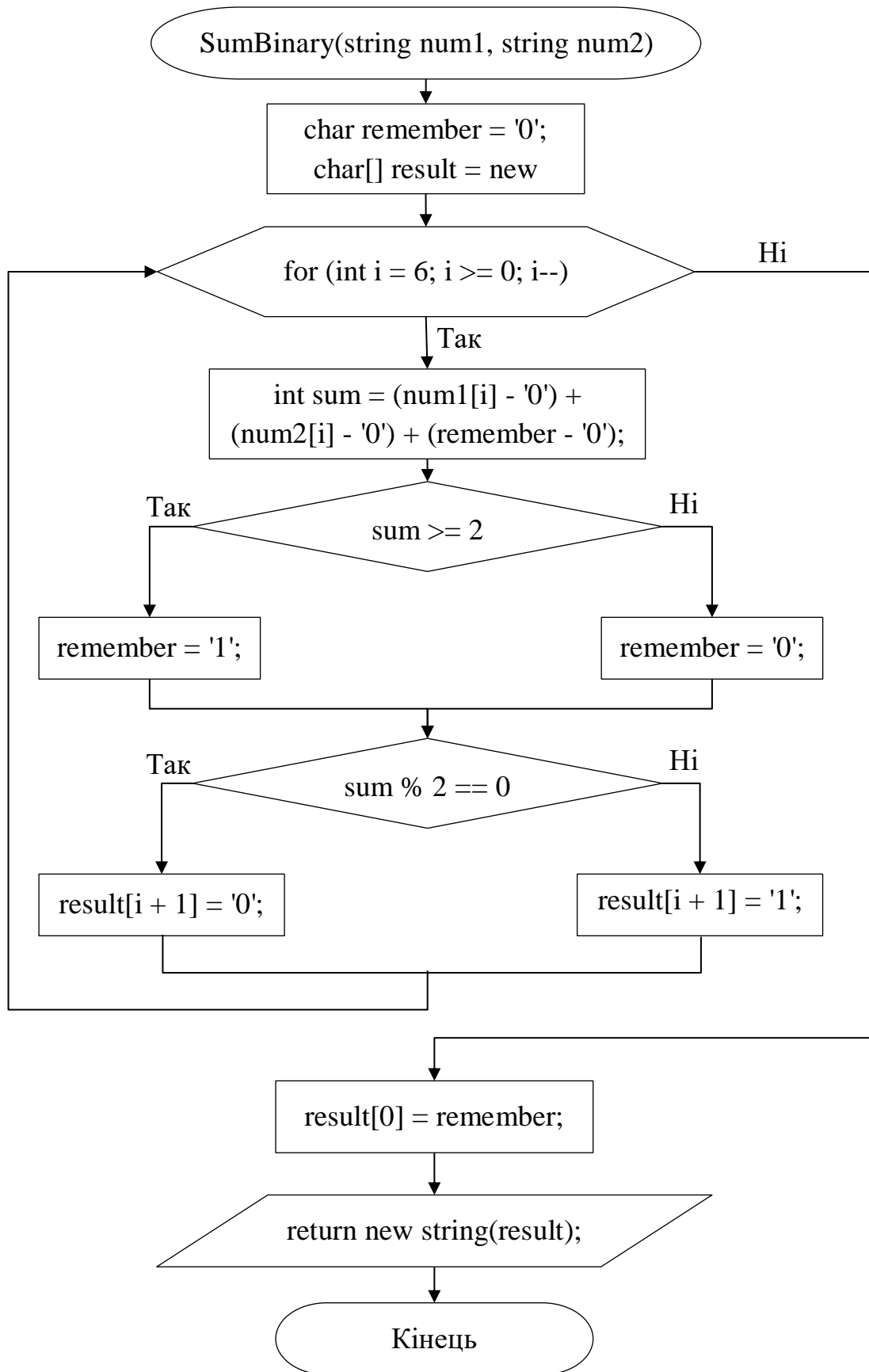


Рис. 1.3. Метод SumBinary()

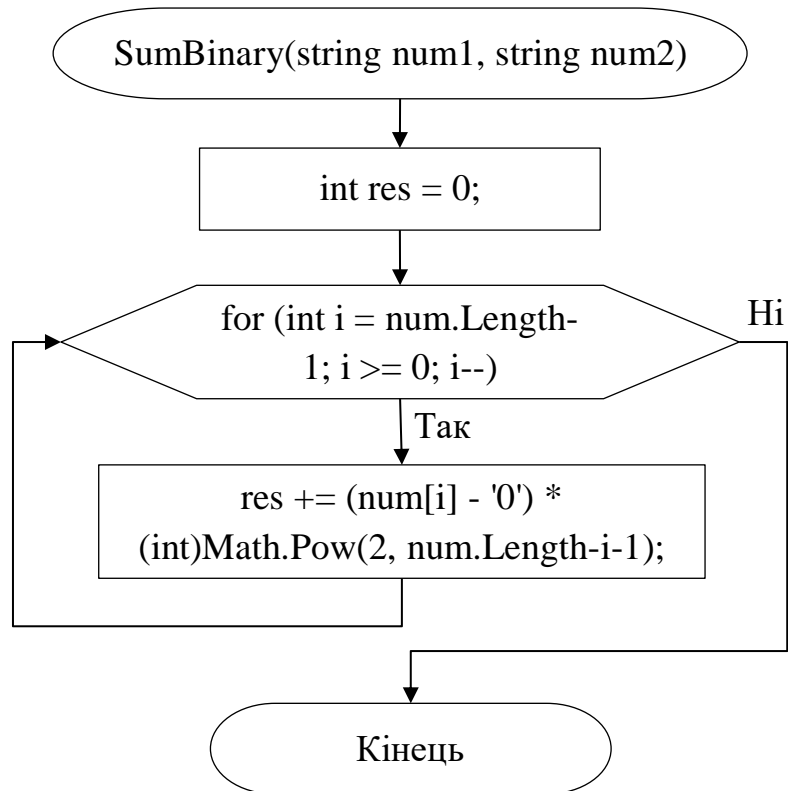


Рис. 1.4. Метод Decimal()

## ЛІСТИНГ ПРОГРАМИ

### GitHub

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading;
using System.Threading.Tasks;

```

```

namespace Lab1
{
    internal class Program
    {
        static void Main()
        {

```

```
        Console.OutputEncoding = UTF8Encoding.UTF8;
        Console.WriteLine("Варіант 2");
        string start;
        do
        {
            Console.WriteLine("\nДля виконання
програми введіть 1");
            Console.WriteLine("Для виходу з програми
введіть 0");

            start = Console.ReadLine();
            switch (start)
            {
                case "1":
                    Console.WriteLine("\nВиконую
програму");

                    Start();
                    break;
                case "0":
                    break;
                default:
                    Console.WriteLine("Команда
''{0}'' не розпізнана. Зробіть, будь ласка, вибір із 1,
0.", start);

                    break;
            }
        } while (start != "0");
    }

    static void Start()
    {
        bool choice;
```

```
        string num1, num2;
        do
        {
            Console.Write("\nВведіть перше число в
бінарному вигляді: ");
            num1 = Console.ReadLine();
            choice = false;
            Check(num1, ref choice);
        }
        while (choice);
        do
        {
            Console.Write("\nВведіть друге число в
бінарному вигляді: ");
            num2 = Console.ReadLine();
            choice = false;
            Check(num2, ref choice);
        }
        while (choice);

        string sumbinary = SumBinary(num1, num2);
        int num1_decimal = Decimal(num1);
        int num2_decimal = Decimal(num2);
        int sumdecimal = num1_decimal + num2_decimal;

        Console.WriteLine($"{num1} + {num2} =
{sumbinary}");
        Console.WriteLine($"{num1_decimal} +
{num2_decimal} = {sumdecimal}");
        if (Decimal(sumbinary) == sumdecimal)
```



```

        Console.WriteLine($"{sumbinary} =
{sumdecimal}\n\nОтже, обчислення виконано правильно.");
    else
        Console.WriteLine($"{sumbinary} !=
{sumdecimal}\n\nПомилка! Обчислення виконано
неправильно.");
    }
    static void Check(string number, ref bool choice)
    {
        foreach (char num in number)
        {
            if (num != '0' && num != '1')
            {
                Console.WriteLine("Помилка! Введене
число не бінарне! Повторіть спробу.");
                choice = true;
                break;
            }
        }
        if (number.Length < 7)
        {
            Console.WriteLine("Помилка! Введене число
складається з меншої кількості цифр! Повторіть спробу.");
            choice = true;
        }
        else if (number.Length > 7)
        {
            Console.WriteLine("Помилка! Введене число
складається з більшої кількості цифр! Повторіть
спробу.");
        }
    }
}

```

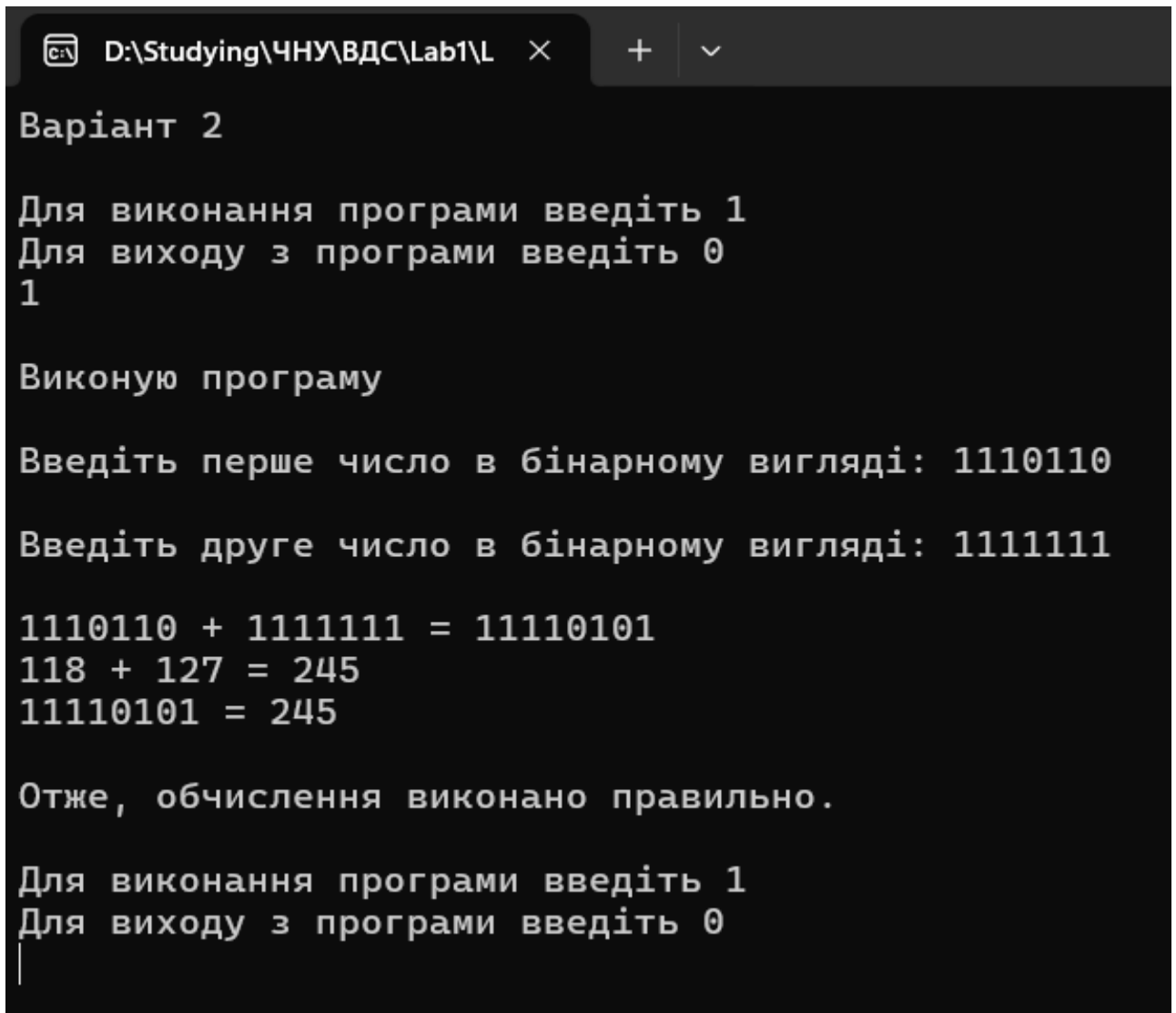
```

        choice = true;
    }
}
static string SumBinary(string num1, string num2)
{
    char remember = '0';
    char[] result = new char[8];
    for (int i = 6; i >= 0; i--)
    {
        int sum = (num1[i] - '0') + (num2[i] -
'0') + (remember - '0');
        if (sum >= 2) remember = '1';
        else remember = '0';
        result[i + 1] = (sum % 2 == 0) ? '0' :
'1';
    }
    result[0] = remember;
    return new string(result);
}
static int Decimal(string num)
{
    int res = 0;
    for (int i = num.Length-1; i >= 0; i--)
        res += (num[i] - '0') * (int)Math.Pow(2,
num.Length-i-1);
    return res;
}
}

```

## РЕЗУЛЬТАТ РОБОТИ ПРОГРАМИ

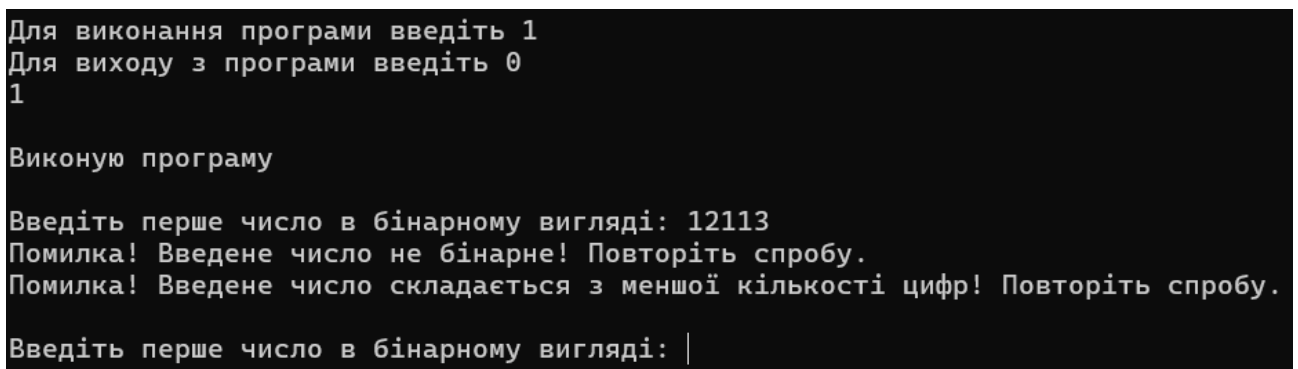
1. Виконання програми (Рис. 2.1);



```
D:\Studying\ЧНУ\ВДС\Lab1\L × + ▾  
Варіант 2  
Для виконання програми введіть 1  
Для виходу з програми введіть 0  
1  
Виконую програму  
Введіть перше число в бінарному вигляді: 1110110  
Введіть друге число в бінарному вигляді: 1111111  
1110110 + 1111111 = 11110101  
118 + 127 = 245  
11110101 = 245  
Отже, обчислення виконано правильно.  
Для виконання програми введіть 1  
Для виходу з програми введіть 0  
|
```

Рис 2.1. Приклад виконання програми в консолі.

2. Виконання програми при помилках у введенні (Рис. 2.2);



```
Для виконання програми введіть 1  
Для виходу з програми введіть 0  
1  
Виконую програму  
Введіть перше число в бінарному вигляді: 12113  
Помилка! Введене число не бінарне! Повторіть спробу.  
Помилка! Введене число складається з меншої кількості цифр! Повторіть спробу.  
Введіть перше число в бінарному вигляді: |
```

Рис. 2.2. Приклад виконання програми в консолі при помилках у введенні.

## **ВИСНОВОК**

У ході виконання лабораторної роботи було розроблено програму для додавання двох бінарних чисел, що складаються з 7 символів. Програма реалізує перевірку введених даних, забезпечуючи коректний ввід лише бінарних чисел заданої довжини. Алгоритм обчислення використовує покрокове додавання з урахуванням перенесення, а отриманий результат виводиться в бінарному та десятковому вигляді, перевіряючи правильність обчислення.

Отримані результати підтвердили правильність реалізації алгоритму. Програма коректно обробляє введені користувачем значення, повідомляючи про помилки при некоректному вводі. Всі розрахунки були перевірені та дали очікуваний результат, що свідчить про правильність роботи алгоритму.