

Міністерство освіти і науки України
ЧЕРКАСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ім. Богдана Хмельницького

Факультет Обчислювальної техніки, інтелектуальних та управляючих систем

Кафедра Програмного забезпечення автоматизованих систем

ЛАБОРАТОРНА РОБОТА № 3

по дисципліні «Програмування та алгоритмічні мови»

Тема: Використання операторів циклу та вибору мови C#

Варіант 6

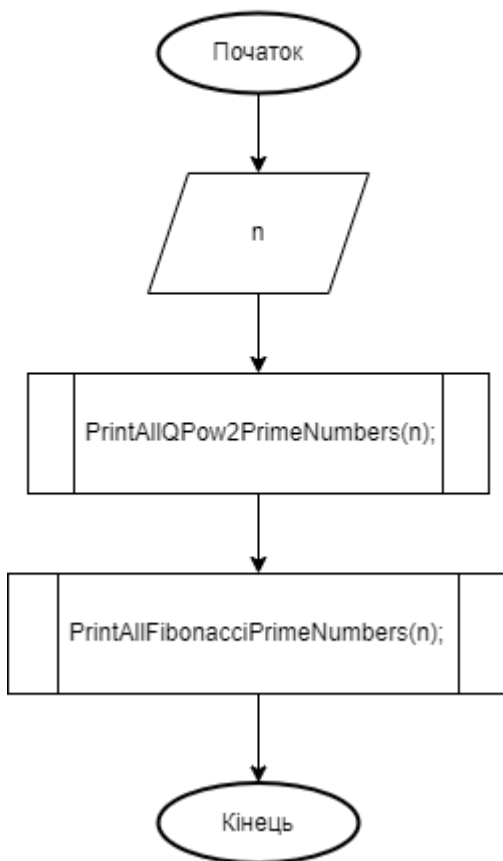
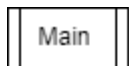
Виконав: студент гр.
КС-231 Киба Д.В.

Черкаси, 2023

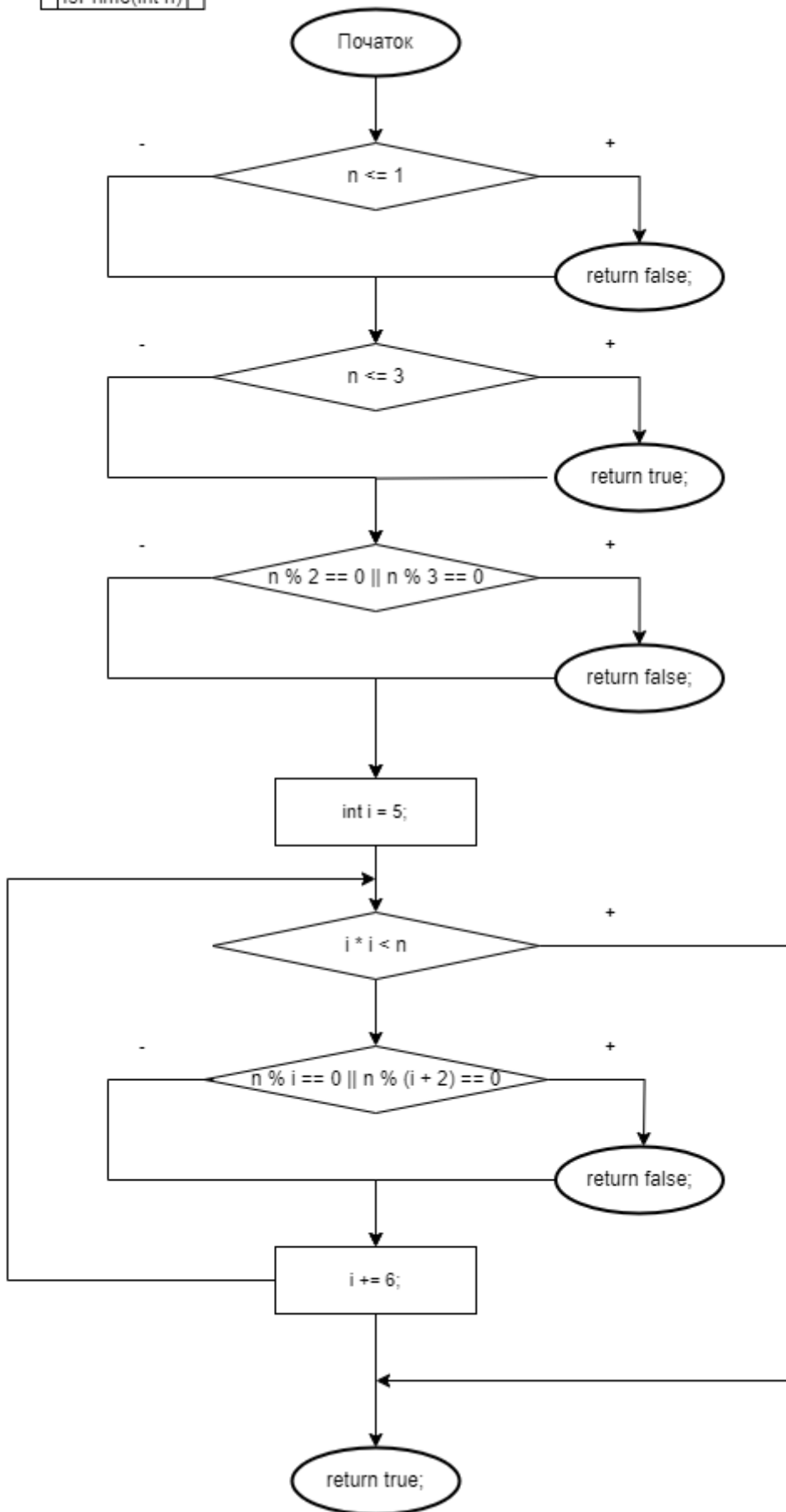
1.1 Постановка завдання :

6. Скласти програму для знаходження всіх чисел із діапазону $[2, n]$, які одночасно є простими числами та числами вигляду $q^2 + 1$, (де q ціле), а також (окремим переліком) всіх чисел із діапазону $[2, n]$, які одночасно є простими числами та числами Фібоначчі. Всього повинно бути три функції (методи): IsPrime, PrintAllQPow2p1PrimeNumbers, PrintAllFibonacciPrimeNumbers. (Звісно, крім них повинна бути також і функція (метод) Main.)

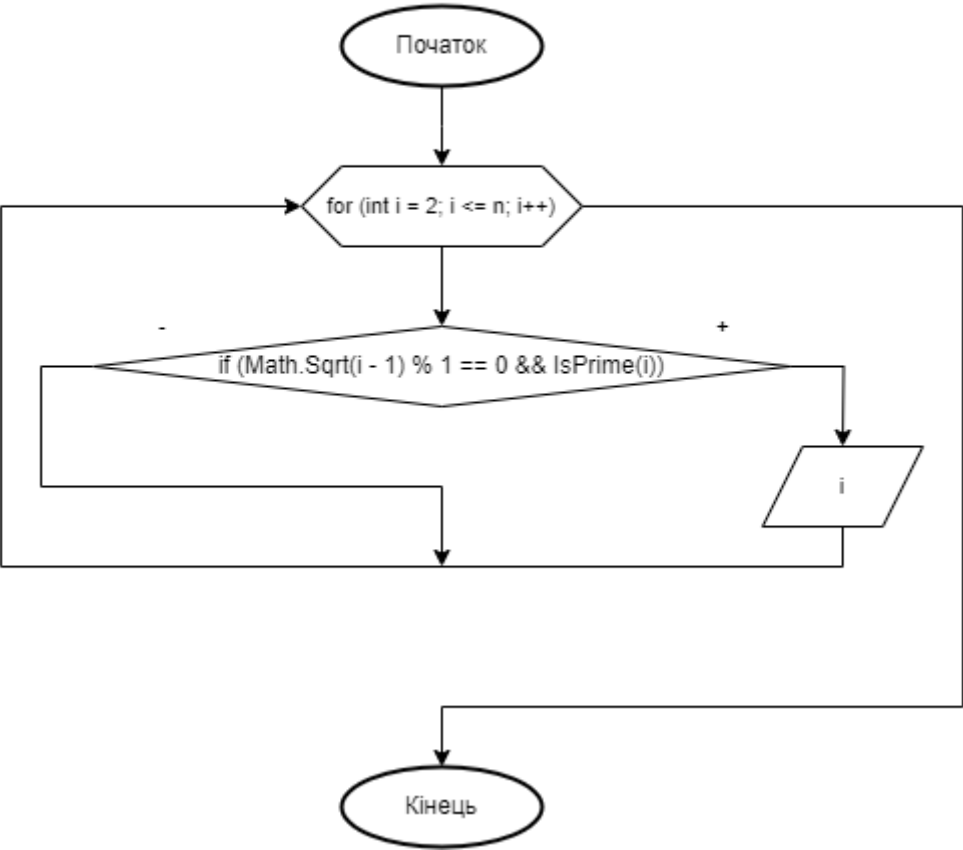
1.2 Алгоритм вирішення завдання:



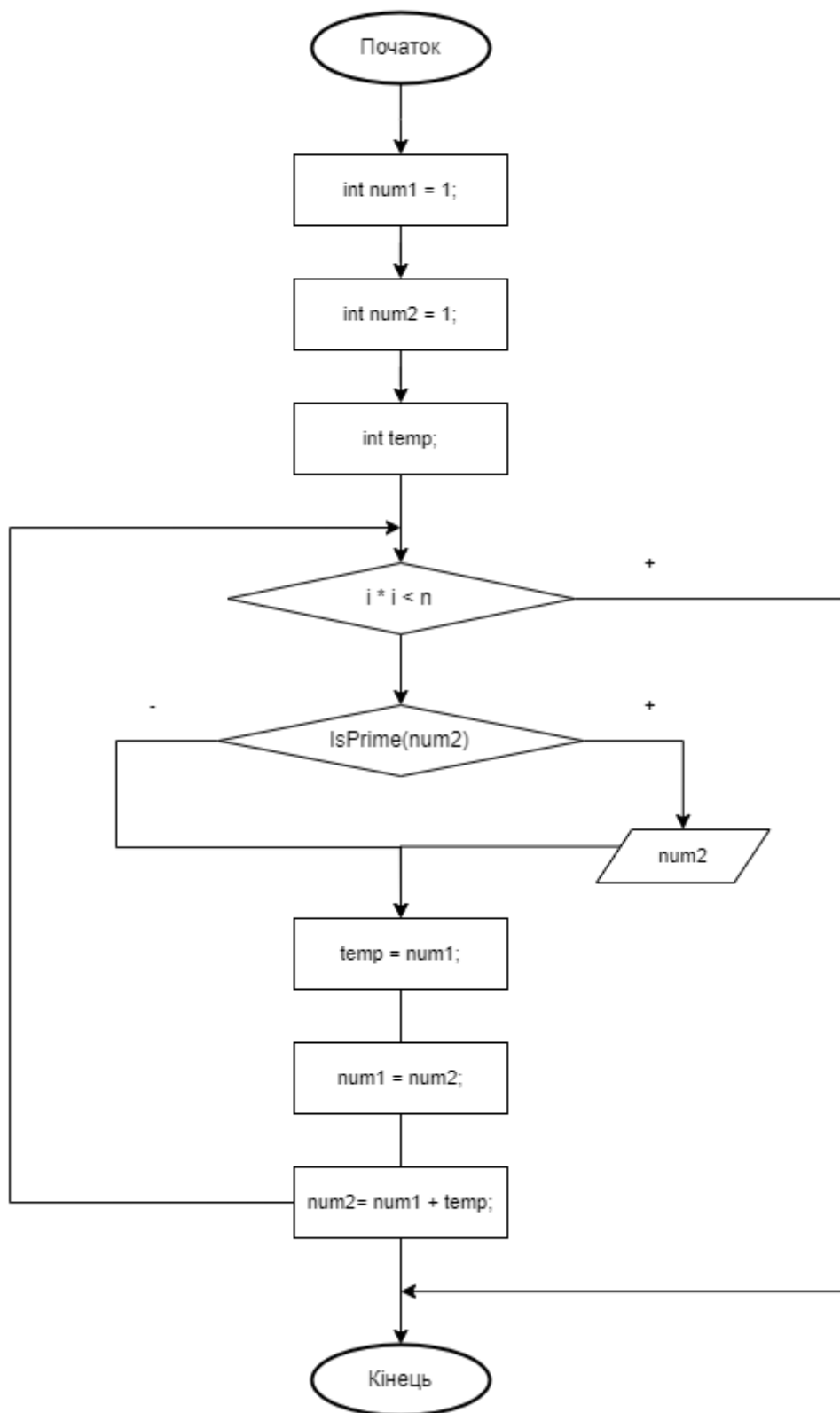
static bool	IsPrime(int n)
-------------	----------------



```
static void  
PrintAllQPow2PrimeNumbers(int n)
```



	static void PrintAllFibonacciPrimeNumbers(int n)	
--	---	--



1.3 Текст програми для вирішення завдань:

```
using System;

using System.Text;

namespace Program
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.OutputEncoding = UTF8Encoding.UTF8;
            System.Console.WriteLine("Введіть значення n: ");
            int n = Int32.Parse(Console.ReadLine());
            System.Console.WriteLine($"Виводжу прості числа вигляду  $q^2 + 1$  від 2 до {n}:");
            PrintAllQPow2PrimeNumbers(n);
            System.Console.WriteLine();
            System.Console.WriteLine($"Виводжу прості числа Фібоначчі від 2 до {n}:");
            PrintAllFibonacciPrimeNumbers(n);
        }

        static bool IsPrime(int n)
        {
            if (n ≤ 1)
            {
                return false;
            }

            if (n ≤ 3)
```

```

    {
        return true;
    }
    if (n % 2 == 0 || n % 3 == 0)
    {
        return false;
    }
    int i = 5;
    while (i * i < n)
    {
        if (n % i == 0 || n % (i + 2) == 0)
        {
            return false;
        }
        i += 6;
    }

    return true;
}

static void PrintAllQPow2PrimeNumbers(int n)
{
    for (int i = 2; i ≤ n; i++)
    {
        if (Math.Sqrt(i - 1) % 1 == 0 && IsPrime(i))
        {
            System.Console.Write($"{i} ");
        }
    }
}

```

```
}
```

```
static void PrintAllFibonacciPrimeNumbers(int n)
```

```
{
```

```
    int num1 = 1;
```

```
    int num2 = 1;
```

```
    int temp;
```

```
    while (num2 ≤ n)
```

```
    {
```

```
        if (IsPrime(num2))
```

```
        {
```

```
            System.Console.Write($"{num2} ");
```

```
        }
```

```
        temp = num1;
```

```
        num1 = num2;
```

```
        num2 = num1 + temp;
```

```
    }
```

```
}
```

```
}
```

```
}
```


1.4 Текст програми для вирішення завдань мовою Golang (додаткові бали):

```
package main

import (
    "fmt"
    "math"
)

func main() {

    fmt.Println("Введіть значення n: ")
    var n int
    fmt.Scan(&n)
    fmt.Printf("Виводжу прості числа вигляду  $q^2 + 1$  від 2 до %d:\n", n)
    print_all_Q_Pow_2_prime_numbers(n)
    fmt.Println()
    fmt.Printf("Виводжу прості числа Фібоначчі від 2 до %d:\n", n)
    print_all_fibonacci_prime_numbers(n)
}

func is_prime(number int) bool {
    if number ≤ 1 {
        return false
    }
    if number ≤ 3 {
        return true
    }
}
```

```

if number%2 == 0 || number%3 == 0 {
    return false
}

i := 5
for i*i ≤ number {
    if number%i == 0 || number%(i+2) == 0 {
        return false
    }
    i += 6
}

return true
}

func print_all_Q_Pow_2_prime_numbers(n int) {
    for i := 2; i ≤ n; i++ {
        if math.Mod(math.Sqrt(float64(i-1)), 1) == 0 && is_prime(i) {

            fmt.Printf("%d ", i)
        }
    }
}

func print_all_fibonacci_prime_numbers(n int) {
    num1 := 1
    num2 := 1

```

```

var temp int

for num2 ≤ n {
    if is_prime(num2) {
        fmt.Printf("%d ", num2)
    }

    temp = num1
    num1 = num2
    num2 = num1 + temp
}
}

```

1.5 Опис формату вхідних даних:

1. У метод Main вводиться ціле число n.
2. Це число n передається як аргумент функціям PrintAllQPow2PrimeNumbers та PrintAllFibonacciPrimeNumbers;

1.6 Посилання на текст програми:

<https://github.com/dimakyba/Lab3>

<https://ideone.com/j1gxug> (C#)

<https://ideone.com/WqDFyZ> (Go)

1.7 Приклади вхідних даних та результатів:

Вводиться ціле число n. Виводяться дві послідовності чисел:

- 1) Прості числа виду $q^2 + 1$.
- 2) Прості числа Фібоначчі.

```

user@LAPTOP MINGW64 /d/Documents/GitHub/Lab3 (main)
$ dotnet run
Введіть значення n:
10000
Виводжу прості числа вигляду q^2 + 1 від 2 до 10000:
2 5 17 37 101 197 257 401 577 677 1297 1601 2917 3137 4357 5477 7057 8101 8837
Виводжу прості числа Фібоначчі від 2 до 10000:
2 3 5 13 89 233 1597
user@LAPTOP MINGW64 /d/Documents/GitHub/Lab3 (main)
$ ^C

user@LAPTOP MINGW64 /d/Documents/GitHub/Lab3 (main)
$ ^C

user@LAPTOP MINGW64 /d/Documents/GitHub/Lab3 (main)
$ ^C

user@LAPTOP MINGW64 /d/Documents/GitHub/Lab3 (main)
$ go run main.go
Введіть значення n:
24000
Виводжу прості числа вигляду q^2 + 1 від 2 до 24000:
2 5 17 37 101 197 257 401 577 677 1297 1601 2917 3137 4357 5477 7057 8101 8837 12101 13457 14401 15377 15877 16901 17957 21317 22501
Виводжу прості числа Фібоначчі від 2 до 24000:
2 3 5 13 89 233 1597

```

1.8 Висновок:

У висновку цієї лабораторної роботи, ми успішно оволоділи основними концепціями використання методів у мові програмування C#. Методи використовуються для того, щоб зручно та ефективно використовувати певний код кілька разів без необхідності повторювати його написання.