

Звіт

До лабораторної роботи №3

по дисципліні «Програмування та алгоритмічні мови»

Тема: Використання операторів циклу та вибору мови C#

Виконала: студента гр. КС-231

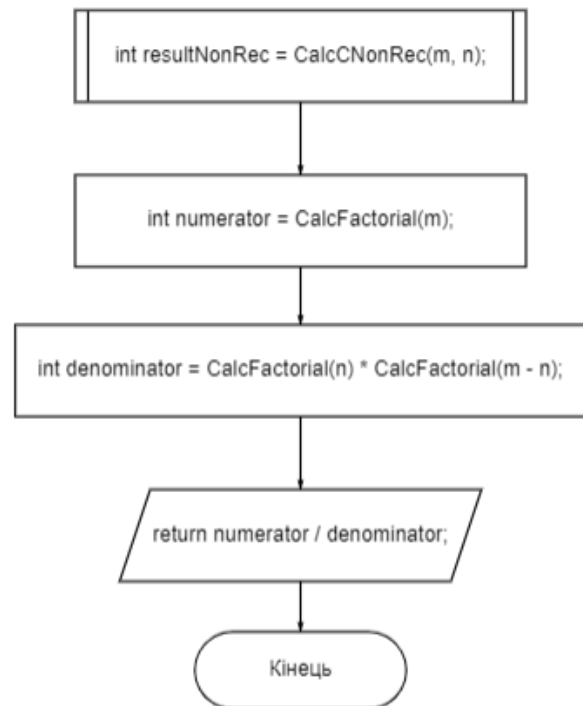
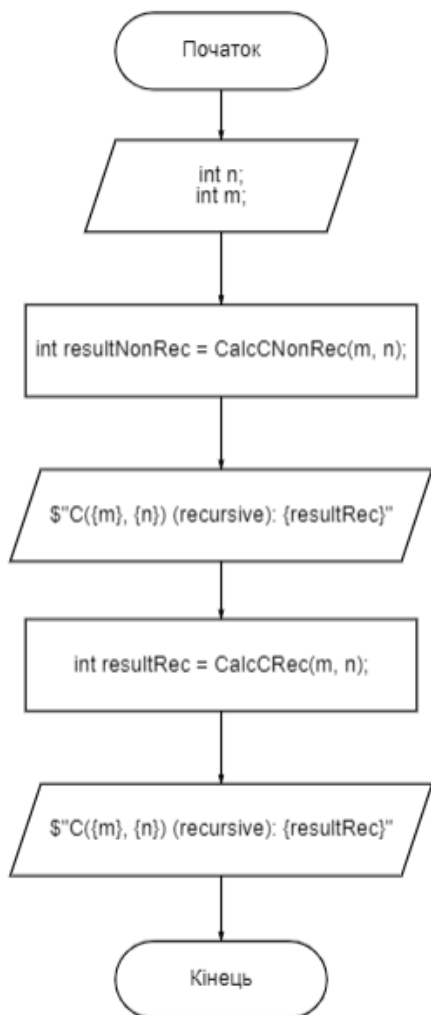
Онищук А.С

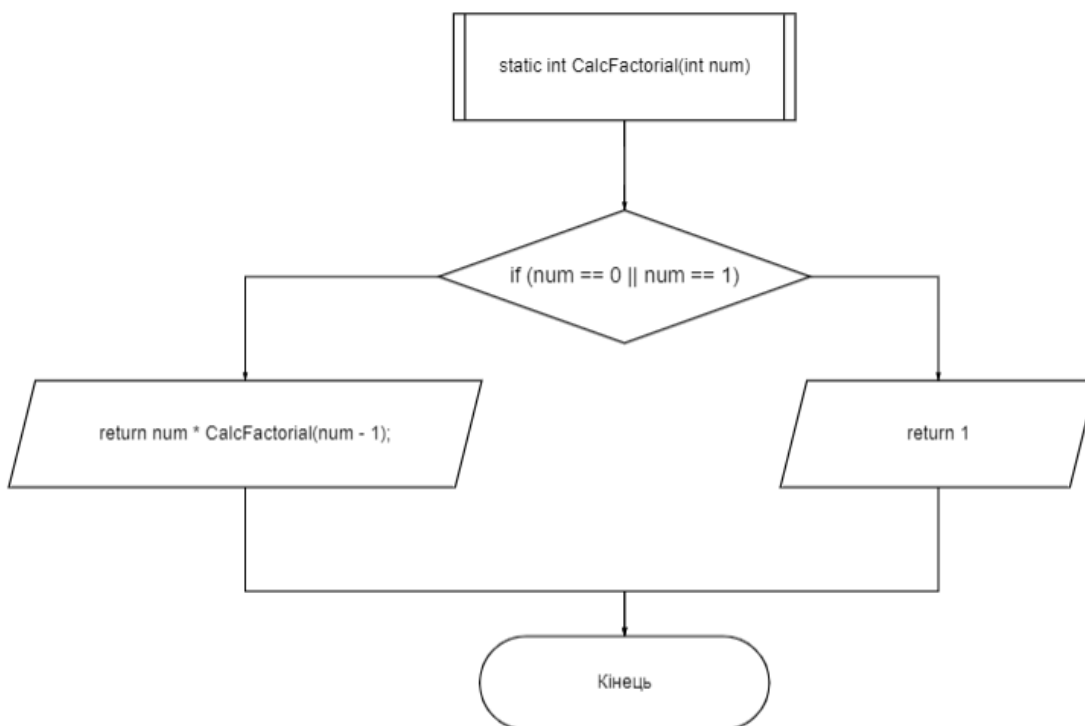
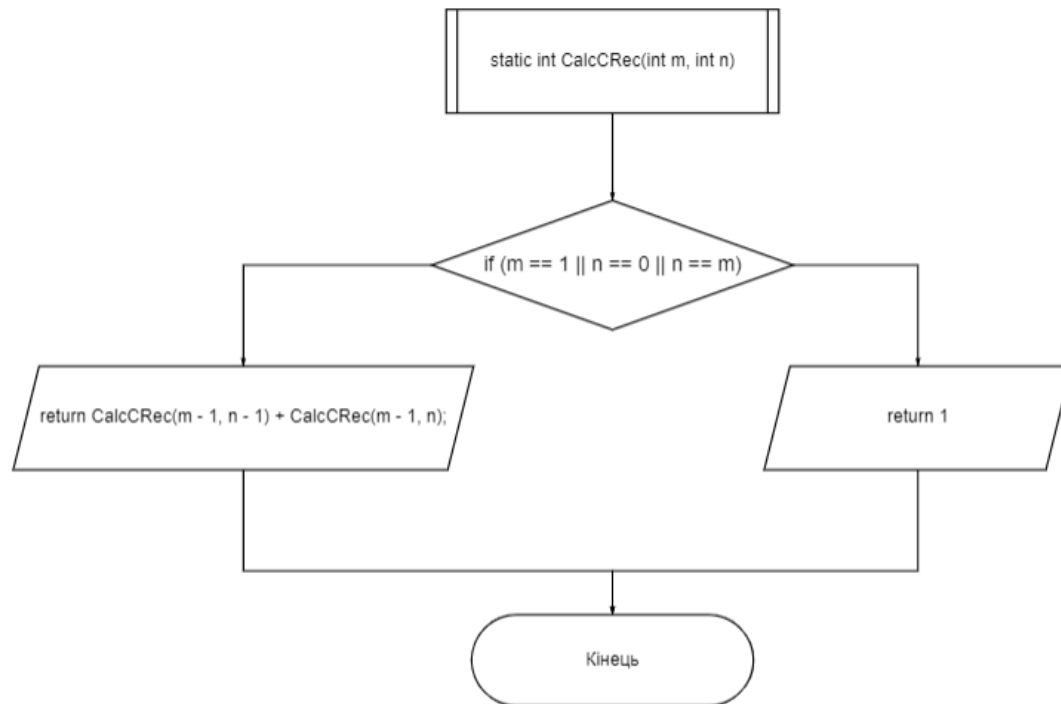
1.1 Постановка завдання:

2. Розрахувати біноміальні коефіцієнти (вони ж кількості сполучень), двома способами: (1) за формулою $C(m, n) = m! / (n! \cdot (m-n)!)$; (2) рекурсивно, через співвідношення: якщо $m \geq 1$, або $n=0$, або $n=m$, то $C(m, n) = 1$; інакше, $C(m, n) = C(m-1, n-1) + C(m-1, n)$. Всього функцій (методів) повинно бути три: CalcCNonRec, CalcFactorial, CalcCRec. (Звісно, крім них повинна бути також і функція (метод) Main. Тобто, перша з цих функцій (методів) CalcCNonRec повинна бути не рекурсивною і використовувати другу функцію (метод) CalcFactorial, а третя функція (метод) CalcCRec повинна обчислювати те саме, що й перша функція (метод) CalcCNonRec, але іншим способом, причому рекурсивно.

1.2 Алгоритм вирішення завдання:

Main:





1.3 Текст програми для вирішення завдань:

```
using System;

class Program
{
    static void Main()
    {
        int m = int.Parse(Console.ReadLine());
        int n = int.Parse(Console.ReadLine());

        int resultNonRec = CalcCNonRec(m, n);
        Console.WriteLine($"C({m}, {n}) (non-recursive): {resultNonRec}");

        int resultRec = CalcCRec(m, n);
        Console.WriteLine($"C({m}, {n}) (recursive): {resultRec}");
        Console.ReadKey();
    }

    static int CalcCNonRec(int m, int n)
    {
        int numerator = CalcFactorial(m);
        int denominator = CalcFactorial(n) * CalcFactorial(m - n);

        return numerator / denominator;
    }

    static int CalcCRec(int m, int n)
    {
        if (m == 1 || n == 0 || n == m)
            return 1;

        return CalcCRec(m - 1, n - 1) + CalcCRec(m - 1, n);
    }

    static int CalcFactorial(int num)
    {
        if (num == 0 || num == 1)
            return 1;

        return num * CalcFactorial(num - 1);
    }
}
```

1.4 Посилання на текст програми:

<https://ideone.com/Jfjuy4>

1.5 Приклади вхідних даних та результатів:

```
F 3
Pr 2
C(3, 2) (non-recursive): 3
C(3, 2) (recursive): 3
```

```
2
3
перше значення не може бути меншим за друге ми їх зам?немо
C(3, 2) (non-recursive): 3
C(3, 2) (recursive): 3
```

1.7 Висновок:

Після виконання цієї лабораторної роботи я отримав важливі навички використання методів у мові програмування C#. Методи стають ефективним інструментом для зручного та повторного використання коду, що полегшує написання програм та робить його більш структурованим.