Додаток 1

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"
Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 4 з дисципліни «Алгоритми та структури даних-1. Основи алгоритмізації»

«Дослідження лінійних алгоритмів»

Варіант **25**

Виконав студент Павленко Микита Андрійович

(шифр, прізвище, ім'я, по батькові)

Перевірив Вечерковська Анастасія Сергіївна

(прізвище, ім'я, по батькові)

Лабораторна робота 6
Дослідження складних рекурсивних алгоритмів

Мета - дослідити особливості роботи рекурсивних алгоритмів та набути практичних навичок їх використання під час складання програмних специфікацій підпрограм.

Варіант 25
Завдання

Отримати всі піфагорові трійки натуральних чисел, кожне з яких не перевищує n, тобто всі такі трійки натуральних чисел a, b, c, що a2+ b2= c2 (a ≤, b ≤, c ≤ n).

1) **Постановка задачі**
   За допомогою підпрограми та рекурсивного процесу знайти всі трійки піфагорових чисел, що не перевищують n.

2) **Побудова математичної моделі**
   Таблиця імен змінних:

Змінна	**Тип**	**Ім'я**	**Призначення**
Границя обчислення	Цілий	**n**	Вхідні дані
Перше число піфагорової трійки	Цілий	**a**	Проміжні дані
Друге число піфагорової трійки	Цілий	**b**	Проміжні дані
Третє число піфагорової трійки	Цілий	**c**	Проміжні дані

Отже, математичне формулювання задачі зводиться до складання логіки підпрограми, реалізації рекурсії та знаходження з їх допомогою трьох значень, що задовольняють заданим умовам.

3) Псевдокод алгоритму

Крок 1:

Функція Recur_a (a, b, c)

Перевірка знайденої трійки чисел

Все функція

початок

Введення **n**

Перебір елементів **c, b**

кінець

Крок 2:

Функція Recur_a (a, b, c)

Якщо (**a > b)** то

Return

Все якщо

Якщо **(a*a + b*b == c*c)** то

Виведення результату

Все якщо

Recur_a(a+1, b, c)

Все функція

початок

Введення **n**

Перебір елементів **c, b**

кінець

Крок 3:

Функція Recur_a (a, b, c)

Якщо (**a > b)** то

Return

Все якщо

Якщо **(a*a + b*b == c*c)** то

Виведення результату

Все якщо

Recur_a(a+1, b, c)

Все функція

початок

Введення **n**

Для **c** від 1 до **n** повторити

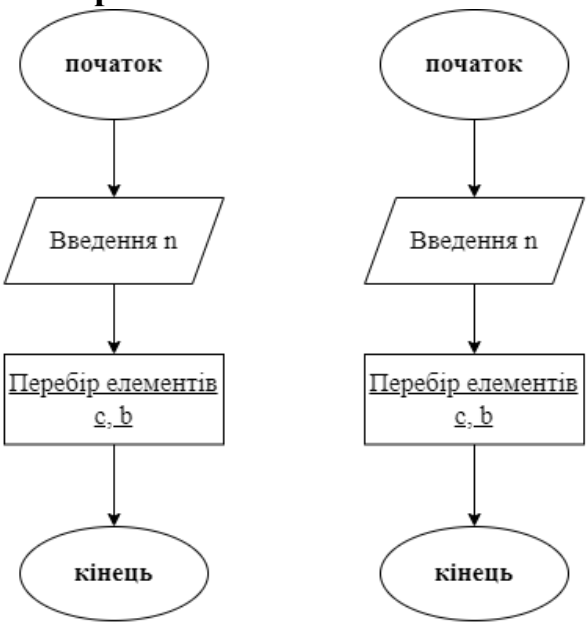
Для **b** від 1 до **n** повторити

Виклик функції **Recur_a(1, b, c)**

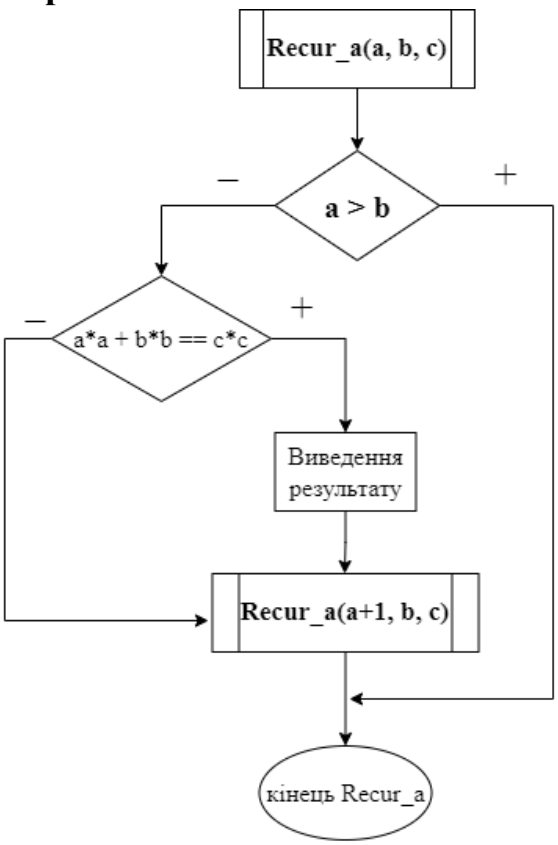
кінець

4) Блок-схема алгоритму

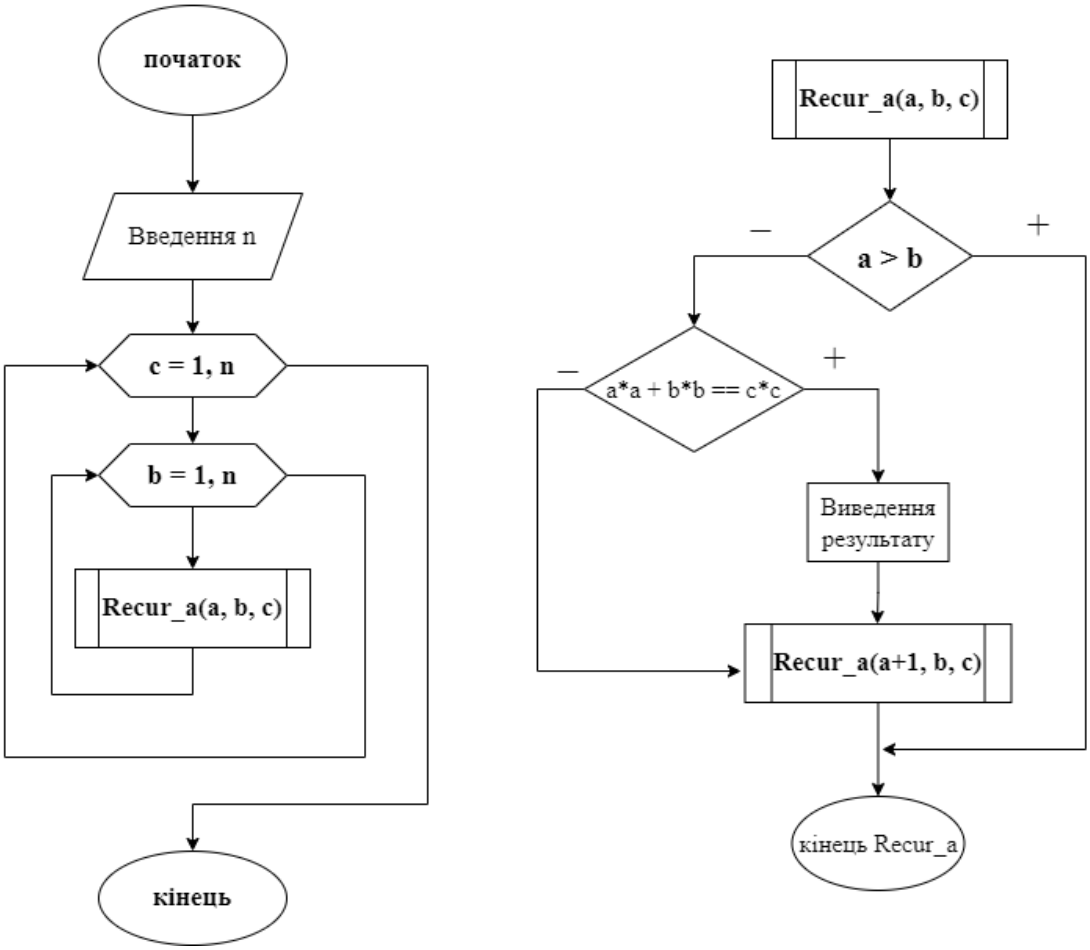
Крок 1



Крок 2



Крок 3



5) Код програми

```
using System;

namespace ConsoleApp4
{
    Ссылок: 0
    class Program
    {
        Ссылок: 2
        static void Recur_a(int a, int b, int c)
        {
            if (a > b) return;
            if (a * a + b * b == c * c)
            {
                Console.WriteLine(a + " " + b + " " + c);
            }
            Recur_a(a+1, b, c);
        }
        Ссылок: 0
        static void Main(string[] args)
        {
            Console.WriteLine("Enter the edge (n)");
            Int32 n = Convert.ToInt32(Console.ReadLine());
            for (int c = 1; c <= n; ++c)
                for (int b = 1; b <= c; ++b)
                    Recur_a(1, b, c);
        }
    }
}
```

Робота з прикладом n = 10

```
Консоль отладки Microsoft Visual Studio
Enter the edge (n)
10
3 4 5
6 8 10
```

Робота з прикладом n = 20

```
Консоль отладки Microsoft Visual Studio
Enter the edge (n)
20
3 4 5
6 8 10
5 12 13
9 12 15
8 15 17
12 16 20
```

5) Випробування алгоритму

Блок	Дія
	Початок
1	Введення: **n** = 10

2	**c = 1** **b = 1** **a = 1**
3	**(a > b)** == false **(a*a + b*b == c*c)** == false **a** = 2
4	**(a > b)** == true
5	**c = 2** **b = 1** **a = 1**
6	**(a > b)** == false **(a*a + b*b == c*c**) == false **a** = 2
7	**(a > b)** == true
8	**b** = 2 **a** = 1
9	**(a > b)** == false **(a*a + b*b == c*c)** == false **a** = 2
10	**(a > b)** == false **(a*a + b*b == c*c)** == false **a** = 3
11	**(a > b)** == true
…	**…**
13	**c = 5** **b = 4** **a = 3**
14	**(a > b)** == false **(a*a + b*b == c*c)** == true
15	Вивід: 3 4 5
…	**…**

16	**c = 10** **b = 8** **a = 6**
17	**(a > b)** == false **(a*a + b*b == c*c)** == true
18	Вивід: 6 8 10
	Кінець

Блок	**Дія**
	Початок
1	Введення: 15
2	Виведення: 3 4 5 6 8 10 5 12 13 9 12 15
	Кінець

Блок	**Дія**
	Початок
1	Введення: 20
2	Виведення: 3 4 5 6 8 10 5 12 13 9 12 15 8 15 17 12 16 20
	Кінець

6) Висновки

Я дослідив особливості роботи рекурсивних алгоритмів та набув практичних навичок їх використання під час складання програмних специфікацій підпрограм. Успішно виконав поставлену задачу.