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 № 6 з дисципліни

«Алгоритми та структури даних-1.

Основи алгоритмізації»

«Дослідження лінійних алгоритмів»

Варіант 25

Виконав студент ІП-14 Радзівіло Валерія Артемівна

(шифр, прізвище, ім'я, по батькові)

Перевірив Мартинова Оксана Петрівна

( прізвище, ім'я, по батькові)

Київ 2021

**Лабораторна робота 6**

**Дослідження рекурсивних алгоритмів**

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

**Задача**:

25. 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).

Користувач задає n, яке має бути більшим за усі числа піфагорової трійки. Після цього на екрані мають з’явитися усі трійки, що відповідають умові a2+ b2= c2. Знаходиться з властивості піфагорових трійок a=m^2-k^2, b=2mk, c=m^2+k^2, де m>k, m,k- натуральні числа.

**Математична модель**

|  |  |  |  |
| --- | --- | --- | --- |
| **Змінна** | **Тип** | **Ім’я** | **Призначення** |
| Функція Pif | void | Pif | Функція |
| Ціле число a | Цілий | a | Частина функції |
| Ціле число b | Цілий | b | Частина функції |
| Ціле число c | Цілий | c | Частина функції |
| Ціле число n | Цілий | n | Частина функції |
| Ціле число m | Цілий | m | Частина функції |
| Ціле число k | Цілий | k | Частина функції |
| Ціле число a1 | Цілий | a1 | Частина функції |
| Ціле число b1 | Цілий | b1 | Частина функції |
| Ціле число c1 | Цілий | c1 | Частина функції |
| Ціле число i | Цілий | i | Частина функції |
| Ціле число n | Цілий | n | Початкове дане |
| Ціле число m | Цілий | m | Початкове дане |
| Ціле число k | Цілий | k | Початкове дане |
| Функція Pif (n,m=2,k=1) | void | Pif | Результат |

Програмні специфікації запишемо у псевдокоді та графічній формі у вигляді блок-схеми.

Крок 1. Визначимо основні дії.

Крок 2. Знаходимо піфагорові трійки.

**Псевдокод**

*крок* 1

**початок**

введення n

обчислення піфагорових трійок, що не перевищують n

виведення знайдених піфагорових трійок

**кінець**

*крок* 2

**початок**

**функція** Pif

**якщо** m>k

**то**

a = pow(m,2) - pow(k,2);

b = 2\*m\*k;

c=pow(m,2)+pow(k,2);

**якщо** a<b && b < c && c <= n

**то**

вивести: a b c

**повторити**

i=2

a1 = a\*i;

b1 = b\*i;

c1=c\*i;

**якщо** c1<=n && (pow(a1,2)+pow(b1,2))== pow(c1,2)

**то**

вивести: a b c

**якщо** c1>n

**то**

**вийти з циклу**

і++

**поки** c1<=n

**якщо** c<=n

**то**

m++;

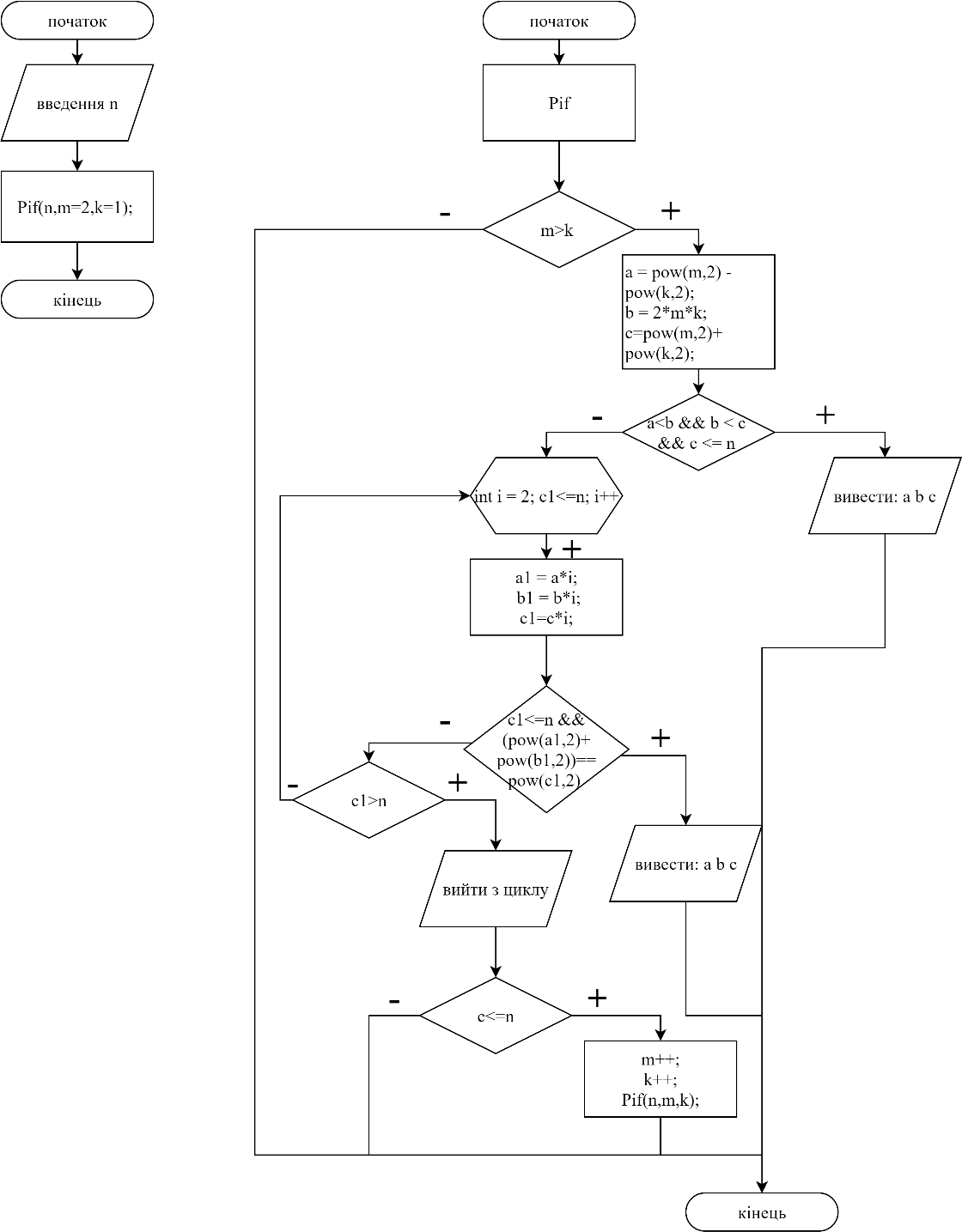
k++;

**виконати те ж,що і в останній раз** Pif(n,m,k);

введення n

Pif(n,m=2,k=1);

**кінець**

**Крок 1**

**Крок 2**

**Код програми С++**

#include <iostream>

#include <cmath>

using namespace std;

//int a = 0,b = 0,c = 0;

void Pif(int n, int &m, int &k)

{

    int a,b,c;

    // a=m^{2}-k^{2},\ b=2mk,\ c=m^{2}+k^{2}}

        if(m>k)

        {

            a = pow(m,2) - pow(k,2);

            b = 2\*m\*k;

            c=pow(m,2)+pow(k,2);

            if(a<b && b < c && c <= n){

            cout << a << " " << b << " "<< c << endl;}

            int c1 = c;

            for (int i = 2; c1<=n; i++)

            {

            int a1 = a\*i;

            int b1 = b\*i;

            int c1=c\*i;

            if(c1<=n && (pow(a1,2)+pow(b1,2))== pow(c1,2)) cout << a1 << " " << b1 << " "<< c1 << endl;

            if(c1>n) break;

            }

                if(c<=n)

                {

                m++;

                k++;

                Pif(n,m,k);

                }

        }

}

int main()

{

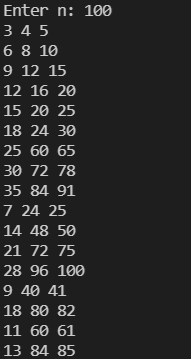
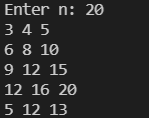
int n,m,k;

cin >> n;

Pif(n,m=2,k=1);

}

**Тестування програми**

**** ****

**Випробовування алгоритму:**

|  |  |
| --- | --- |
| 1. | Введемо дані: n = 10 |
| 2. | Pif(10,2,1) |
| 3. | Ітерації:   1. a = 3 b = 4 c = 5 2. a = 6 b = 8 c = 10 |
| 5. | Вивід: 3 4 5  6 8 10 |

**Висновок**: Було досліджено особливості роботи рекурсивних алгоритмів та набути практичних навичок їх використання під час складання програмних специфікацій підпрограм. У алгоритмі були уточнені обмеження з використанням умов при яких задача буде виконана правильно, а також функція, що визначає Піфагорові трійки, кожне число з яких не перевищує задане користувачем n. За допомоги рекурсії визначені значення цих трійок з обмеженнями. За допомоги цієї лабораторної роботи можливо обчислення Піфагорових трійок.