Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту

A blue and white logo

Description automatically generated

**Звіт**

**Звіт**

**про виконання лабораторних та практичних робіт блоку № 2**

На тему: «Програмування: алгоритм, програма, код. Системи числення. Двійкова система числення. Розробка та середовище розробки програми.»

***з дисципліни:*** «Основи програмування»

до:

Практичних Робіт до блоку № 2

**Виконав:**

Студент групи ШІ-13

Литвин Маркіян Назарович

Львів 2024

**Тема:** Лінійні алгоритми. Розгалужені алгоритми. Умовні та логічні оператори. Змінні. Константи. Типи даних. Розмір типів даних. Ввід вивід. Базові операції та вбудовані функції. Коментарі.

**Мета:** навчитися ефективно будувати програмну логіку та працювати з різними типами дани, створювати чіткі й структуровані програми, що вирішують різноманітні задачі з використанням стандартних інструментів програмування.

**Теоретичні відомості:**

1. Системи числення:

○ Визначення та види систем числення (двійкова, вісімкова, десяткова, шістнадцяткова).

○ Перетворення чисел між системами.

○ Практичне застосування систем числення в програмуванні.

○ Вправи на перетворення чисел між системами.

2. Компіляція:

○ Етапи компіляції.

○ Препроцесор і директива include.

○ Функції компілятора.

3. Змінні, Константи, Типи даних та їх Розміри:

○ Визначення та оголошення змінних і констант.

○ Основні типи даних у С++ (int, char, float, double, тощо).

○ Розміри типів даних і їх представлення у двійковій системі.

○ Порівняння різних типів даних та вибір відповідного типу для задач.

4. Бібліотеки в C++

○ Поняття бібліотеки;

○ Використання бібліотеки;

○ Бібліотеки C++ - stdio, cmath, iostream (Стандартна бібліотека C++)

5. Ввід та Вивід даних:

○ Основи використання cin та cout.

○ Основи використання printf/scanf.

○ Форматування виводу даних.

○ Робота з різними типами даних у вводі/виводі.

○ Практичні вправи на ввід та вивід даних.

6. Базові Операції та Вбудовані Функції:

○ Арифметичні операції та їх використання.

○ Побітові операції

○ Використання математичних функцій (sqrt, pow, тощо).

○ Вправи на роботу з базовими операціями та функціями.

7. Коментарі у Коді:

○ Важливість коментарів у програмуванні.

○ Види коментарів у С++.

○ Написання ефективних коментарів.

○ Практика коментування коду.

8. Лінійні алгоритми:

○ Визначення та особливості лінійних алгоритмів.

○ Структура та властивості лінійних алгоритмів.

○ Написання лінійних алгоритмів на С++.

○ Вправи на створення лінійних алгоритмів.

9. Розгалужені алгоритми та Умовні Оператори:

○ Введення в розгалужені алгоритми.

○ Область видимості.

○ Умовні оператори: if, else, else if, switch-case.

○ Тернарний оператор ?.

○ Реалізація розгалужених алгоритмів на С++.

○ Практичні задачі на розгалужені алгоритми.

10. Логічні Оператори:

○ Визначення та види логічних операторів (AND, OR, NOT).

○ Застосування логічних операторів у розгалужених алгоритмах.

○ Практичні приклади використання логічних операторів.

○ Вправи на розуміння та використання логічних операцій.

**Індивідуальний план опрацювання теорії:**

1. Системи числення
2. Компілятор
3. Змінні та типи даних
4. Бібліотеки C++
5. Ввід/вивід
6. Вбудовані функції
7. Коментарі
8. Лінійні алгоритми
9. Розгалужені алгоритми та Умовні Оператори

Логічні Оператори

**Джерела**:

* https://www.youtube.com/watch?v=KymlNuMfcV4
* <https://acode.com.ua/uroki-po-cpp/#toc-0>
* **Chatgpt**
* **Лекційний матеріал**

**Виконання роботи:**

**Завдання №1: VNS Lab 1**

**Варіант 9**

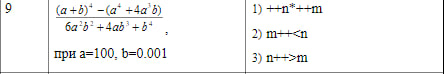
1. Обчислити значення виразу при різних дійсних типах даних (float й double).

Обчислення варто виконувати з використанням проміжних змінних.

Порівняти й пояснити отримані результати.

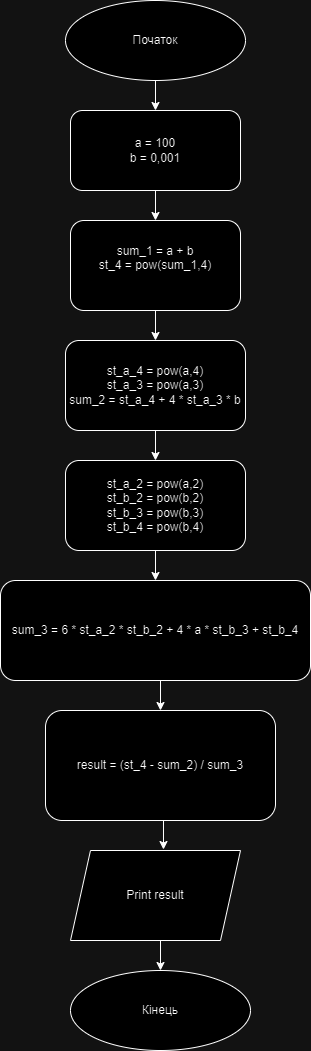
2. Обчислити значення виразів. Пояснити отримані результати.

Task 1 Task 2



**Task 1:**

**Блок-схема:**

****

**Розв’язок:**

****

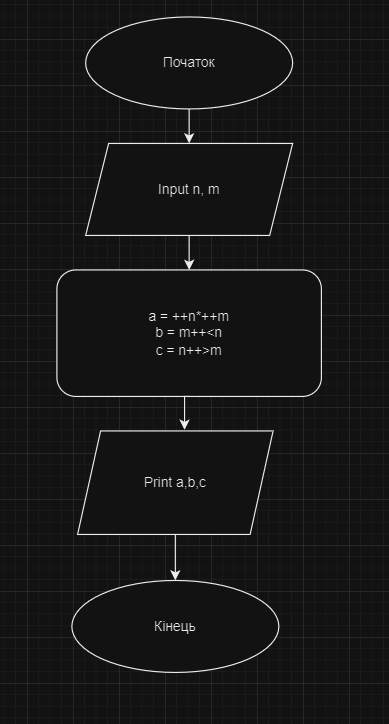
**Результат:**

****

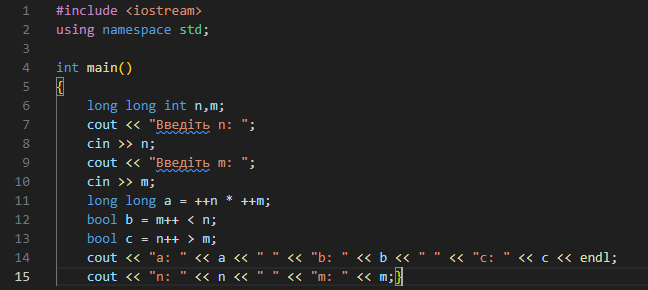
**Час виконання 15 хв**

**Task 2:**

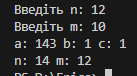
**Блок-схема:**

****

**Розв’язок:**

****

**Результат:**

****

**Час виконання 10 хв.**

**Завдання №2: Algotester Lab 1:**

**Варіант - 1**

У вас є стіл, у якого є 4 ніжки, довжини яких вам дано.

Ви хочете зробити ніжки рівної довжини, для цього ви відпиляєте dd від кожної ніжки (тобто вам буде дано 4 числа, кожне з яких буде означати відпилювання від відповіної ніжки стола).

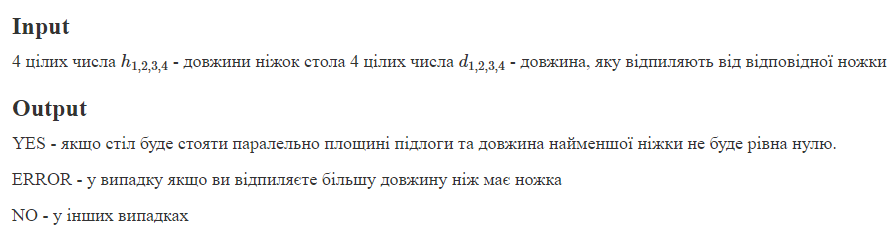
Якщо під час відпилювання найдовша ніжка стола буде у 2 рази більша-рівна ніж найменша ніжка - стіл перевернеться, але відпилювати ніжки це вам не завадить.

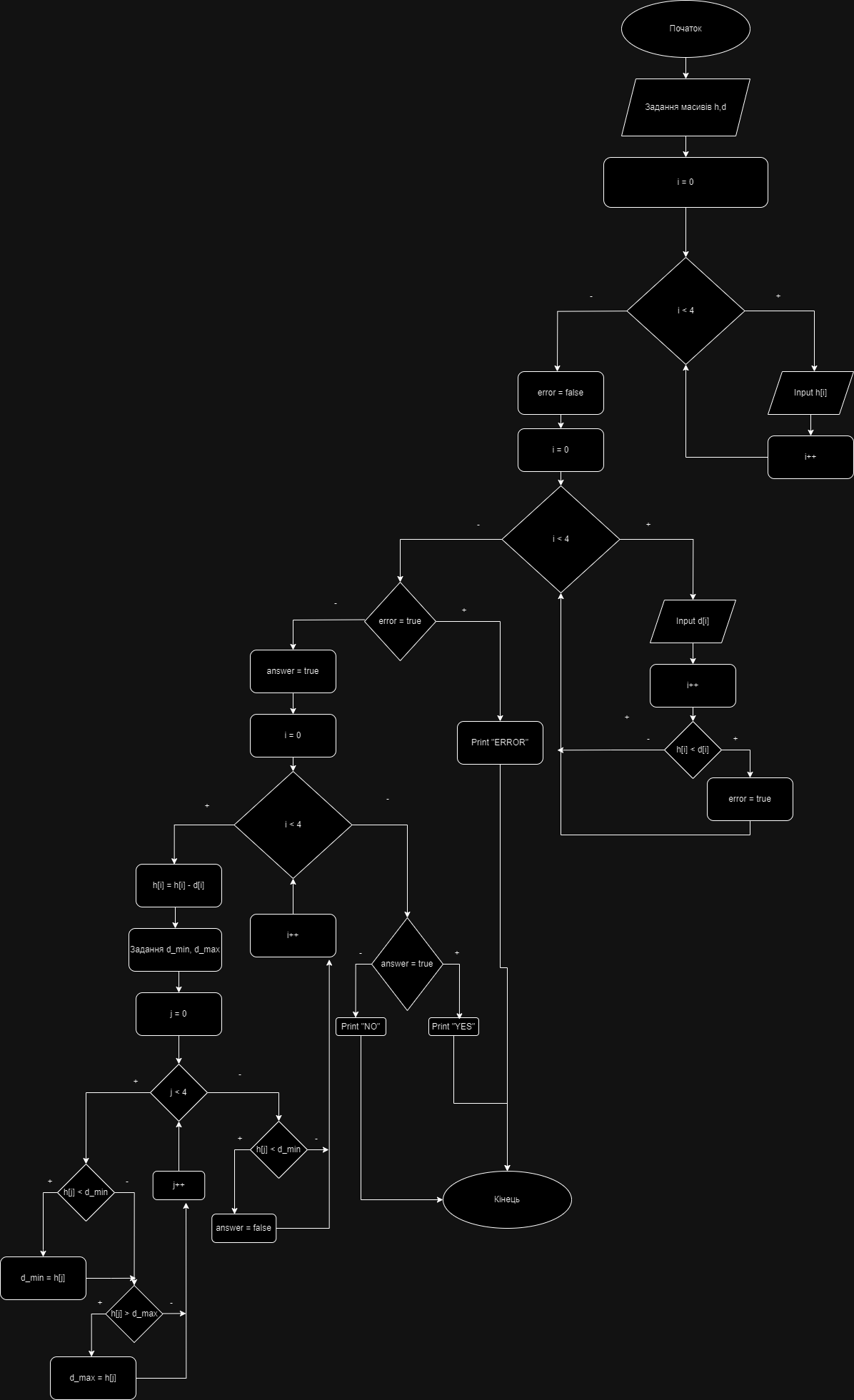
Тобто якщо hmax>=2∗hmin то стіл перевертається. **Увага**, це може статися і між початком та кінцем відпилювання, наприклад коли відпиляють 2, але ще не встигнуть відпиляти 3тю ніжку. Також ми вважаємо що перед відпилюванням стіл не перевернеться.

Ваше завдання сказати чи після усіх маніпуляцій стіл буде цілий та паралельний підлозі.

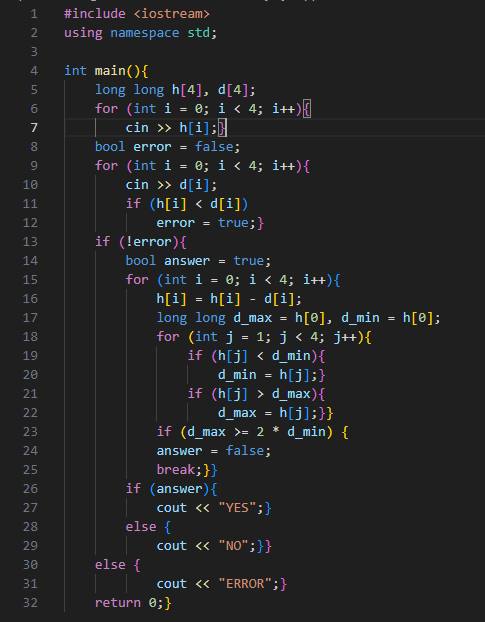
Якщо довжина, яку відріжуть буде більша за довжину ножки - вам треба вивести ERROR.

Увага! Навіть якщо стіл перевернеться - ви все одно відпилюєте ніжки і можете отримати ERROR.

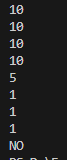
 **Блок-схема:**



**Розв’язок:**

****

**Результат:**

**** ****

**Час виконання: 1.5 години**

**Завдання №3: Practice Task:**

Ви створюєте простий порадник щодо погоди. Користувач вводить поточні погодні умови, а програма видає рекомендації щодо активності на основі погоди.

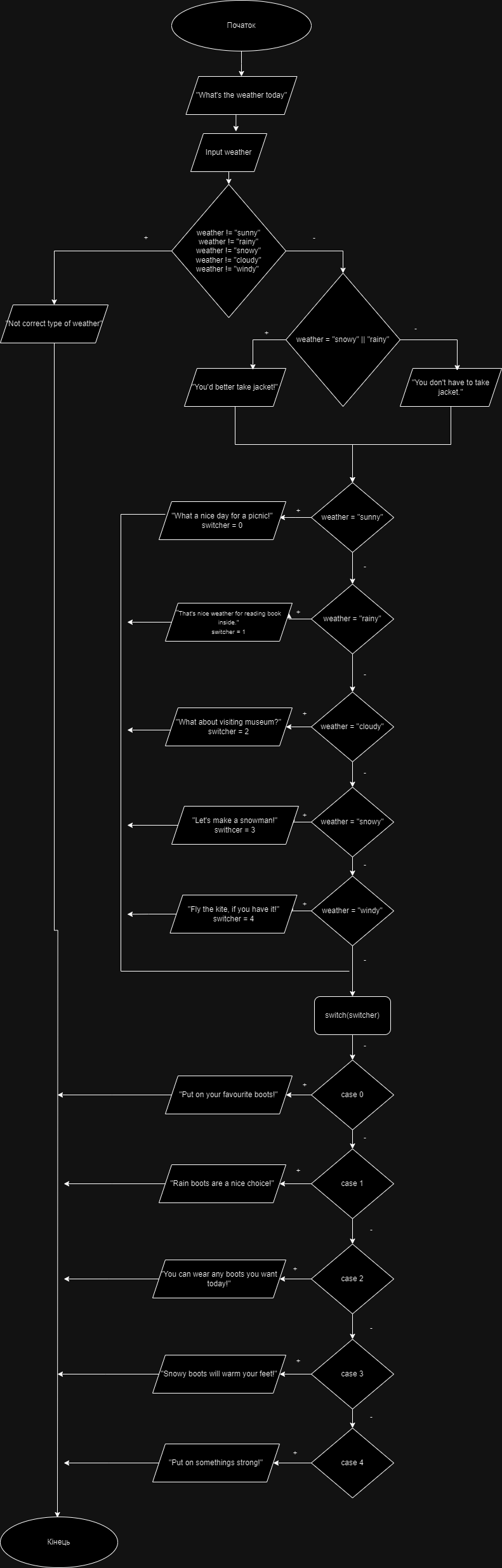
Можливі варіанти погоди:

* sunny;
* rainy;
* cloudy;
* snowy;
* windy;

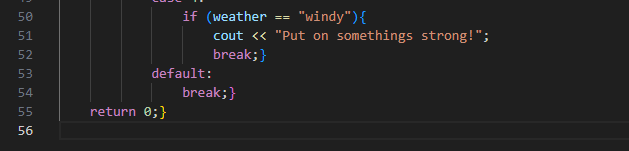
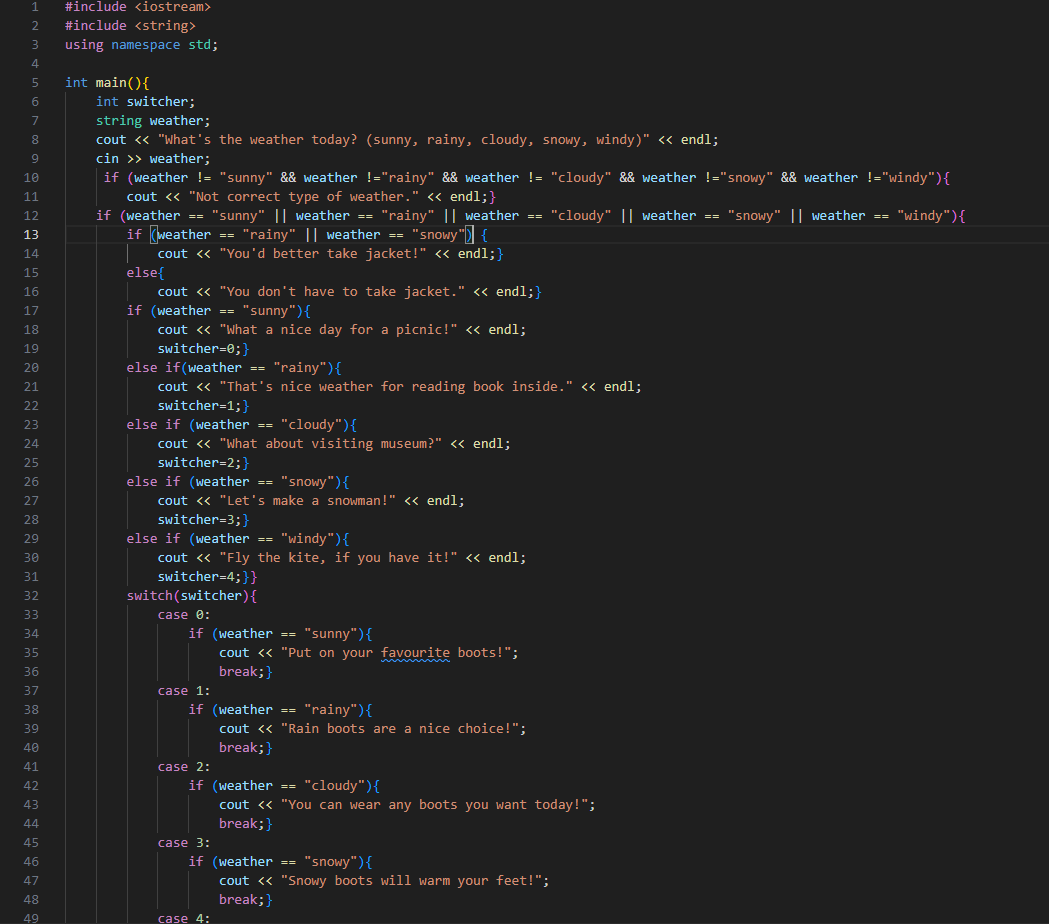
**Вимоги:**

1. Використати всі згадані в передумові задачі оператори галуження - *if else, if, else if, switch case*;
2. За потреби комбінувати оператори;

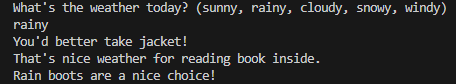
**Блок-схема:**

****

**Розв’язок:**

****

**Результат:**

****

**Час виконання: 35 хв**

**Завдання №4: Self Algotester Task:**

У вашого персонажа є H хітпойнтів та M мани.

Персонаж 3 рази використає закляття, кожне з яких може використати хітпойнти та ману одночасно.

Якщо якесь закляття забирає і хітпойнти і ману - ваш персонаж програє, отже для виграшу треба використовувати при одному заклинанні **АБО** хітпойнти, **АБО** ману.

Якщо в кінці персонаж буде мати додатню кількість хітпойнтів та мани (H,M>0H,M>0) - він виграє, в іншому випадку програє.

Ваше завдання у випадку виграшу персонажа вивести **YES**, вивести **NO** у іншому випадку.

# **Input**

2 цілих числа HH та MM - хітпойнти та мана персонажа

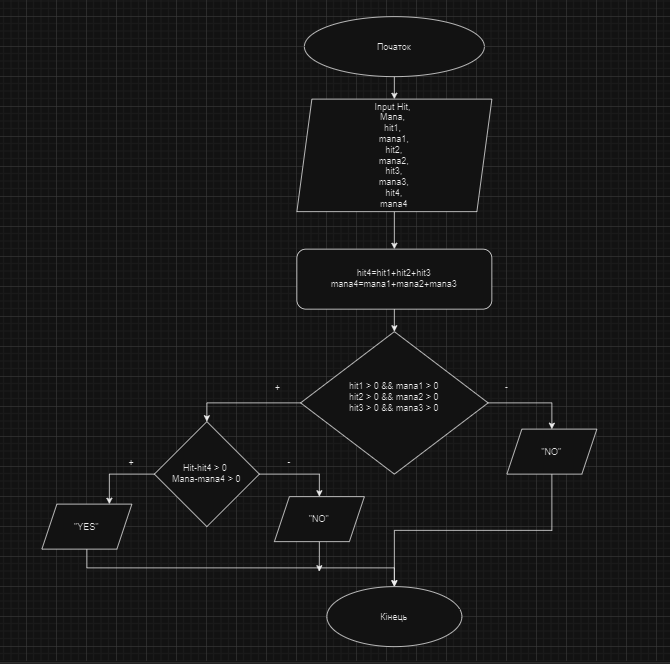
3 рядки по 2 цілих числа, hihi та mimi - кількість хітпойнтів та мани, які ваш персонаж потратить за хід на ii заклинання

# **Output**

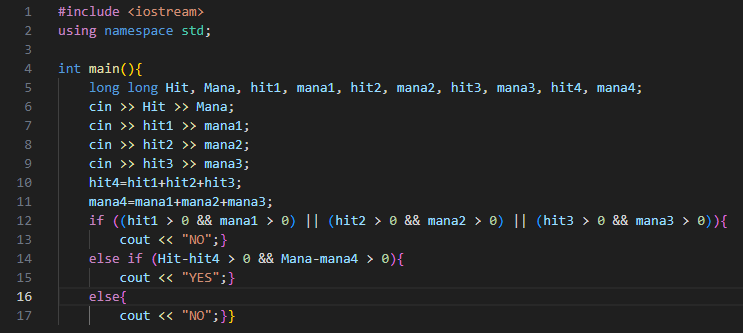
YESYES - якщо ваш персонаж виграє

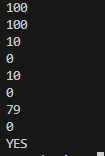
NONO - у всіх інших випадках

**Блок-схема:**

****

**Розв’язок:**

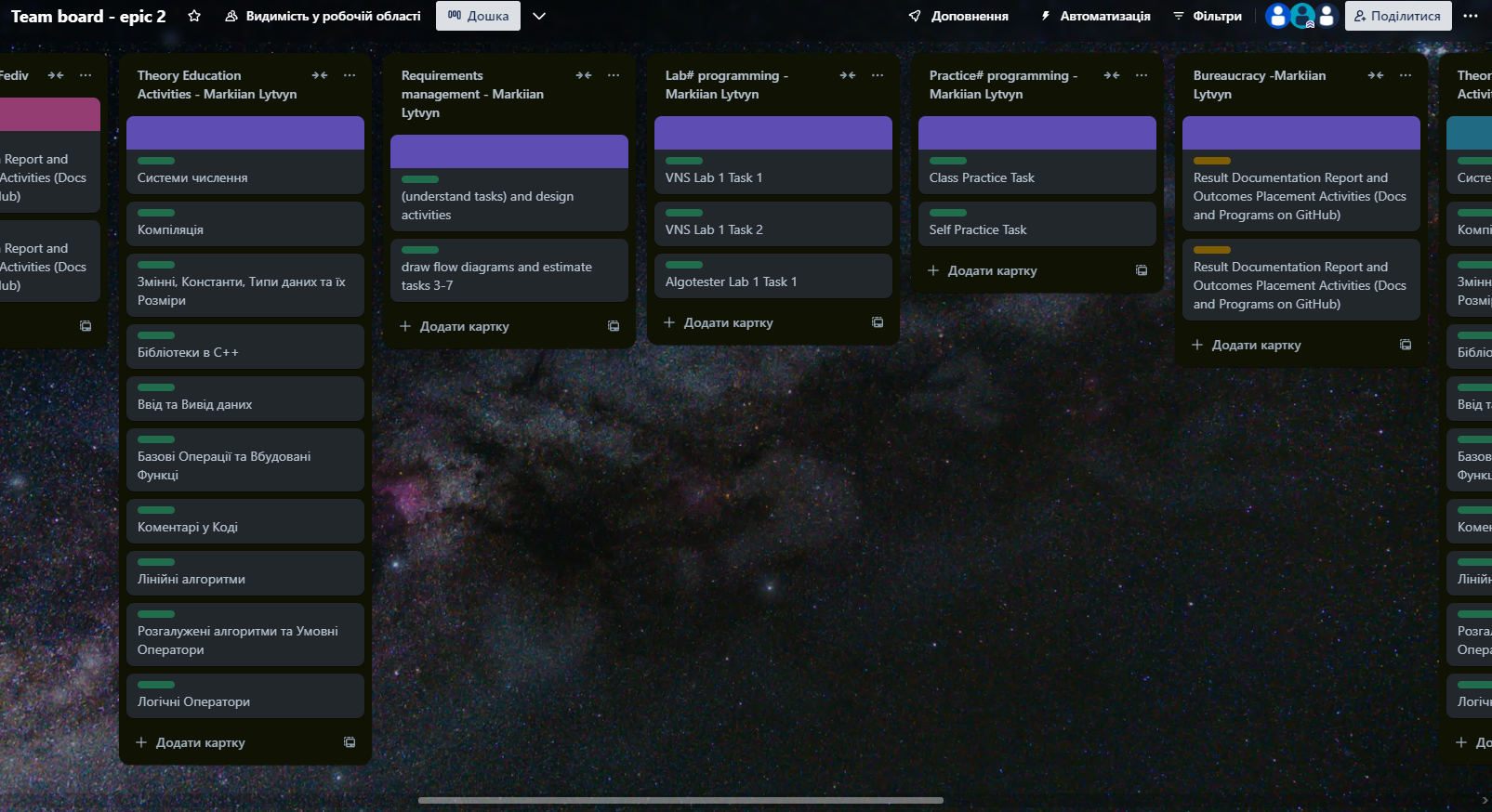
**  
Результат:**

**** ****

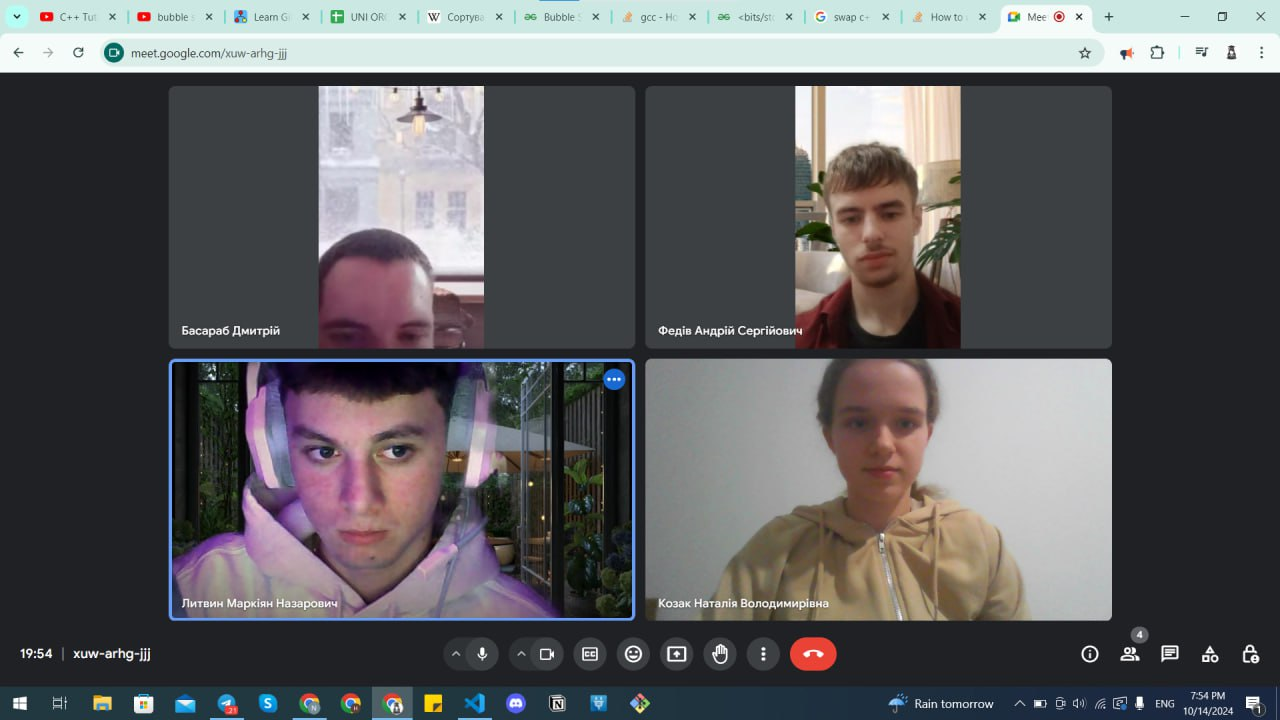
**Час виконання: 25 хв**

**Team work:**

**Trello**



**Зустрічі з командою**



**Висновок:**

Виконуючи цей епік я навчився користуватись операторами if-else та switch, поглибив свої знання з систем числення, компілятору, змінних та типів даних, бібліотеки C++, вводу/виводу, вбудованих функцій, лінійних алгоритмів, розгалужених алгоритмів та операторів.