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

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

до:

Практичних Робіт до блоку № 2

**Виконав:**

Студент групи ШІ-13

Мурашко Владислав Сергійович

Львів 2024

**Тема роботи:**

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

**Мета роботи:**

Навчитися писати лінійні та розгалужені алгоритми, використовувати умовні та логічні оператори, змінні та константи, навчитися різниці між типами даних. Робота з вводом ти виводом даних, коментарями, виконання базових операцій та використання вбудованих функцій.

**Теоретичні відомості:**

1. **Теми, необхідні для виконання роботи:**

* Лінійні та розгалужені алгоритми.
* Умовні та логічні оператори.
* Типи даних, їх розмір. Змінні та константи.
* Ввід та вивід.
* Базові операції та вбудовані функції мови С++.
* Коментарі.

1. **Джерела використані для ознайомлення з вищезазначеними темами:**

* З усією необхідною інформацією я був знайомий з власного досвіду.

**Виконання роботи:**

1. *Опрацювання завдання та вимог до програми та середовища*

**Завдання №1 - Class Practice Work - Особистий порадник**

**Задача**

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;

**Мета Задачі**

Навчитися користуватися операторами галуження для структурування логіки програм.

Якщо користувач вводить будь-яку іншу умову, запропонуйте йому ввести дійсну умову.

**Використовуйте таку логіку**

*if else* - щоб вирішити, чи повинен користувач взяти куртку чи ні.

*if, else if* - щоб надати рекомендацію щодо активності (прогулянка, футбол, настільні ігри, etc).

*switch case* - для визначення типу рекомендованого взуття.

**Завдання №2 - Self Practice Work – Algotester lab 2 variant 3**

**Задача:**

Вам дано масив цілих чисел розміром N, на першій та останній клітинці розміщено по дрону. Вони одночасно взлітають.

На початку кожного ходу швидкість дрону стає рівною значенню клітинки, у якій він знаходиться.

Тобто лівий дрон у першу секунду з клітинки з індексом 1 перелетить у клітинку з індексом a1, тобто його наступна позиція рахується як поточна позиція + число у поточній позиції (перегляньте пояснення для візуалізації) Правий робить аналогічно в протилежну сторону.

Вони це роблять до моменту, коли трапиться одна з зазначених подій:

Якшо 2 дрони опиняються в одній клітинці - ви виводите **Collision**.

Якщо лівий дрон опиниться справа від правого - це **Miss**

У випадку якщо вони зупиняться один навпроти одного, тобто у клітинках ai та ai+1 - виведіть **Stopped**

Врахуйте, що перевіряти треба також до взльоту.

**Constraints**

1≤N≤1000

1≤ai≤5

**Гарантується**, що дрони не можуть вилетіти за межі [1,N] (тобто не існує масиву, в якому дрони вилітають за межі до того, як станеться одна з 3-х вищезазначених ситуацій) та їх швидкість завжди більша за нуль.

**Завдання №3** **– Algotester lab 1 variant 3**

**Задача:**

Персонажу по одному дають сторони 5 кубів a1..5, з яких він будує піраміду.

Коли він отримує куб з ребром ai - він його ставить на існуючий, перший ставить на підлогу (вона безмежна).

Якщо в якийсь момент об’єм куба у руці (який будуть ставити) буде більший ніж у куба на вершині піраміди - персонаж програє і гра закінчується. Розмір усіх наступних кубів після програшу **не враховується**.

Тобто якщо ai−1<ai - це програш.

Ваше завдання - сказати як закінчиться гра.

**Input**

5 цілих чисел a1..a5a1..a5 - сторони кубів

**Output**

Існуючі варіанти:

**LOSS** - якщо персонаж не зможе поставити куб.

**WIN** - якщо персонаж зможе поставити усі куби

**ERROR** - якщо сторона куба буде мати неможливу довжину, тобто ai <= 0

**Constraints**

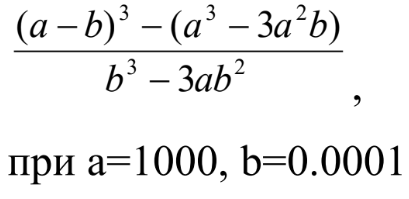
≤ai≤

**Завдання №4 - VNS Lab 1 - Task 1-16**

**Задача:**

Обчислити значення виразу при різних дійсних типах даних (float й double). Обчислення варто виконувати з використанням проміжних змінних.

Порівняти й пояснити отримані результати.



**Завдання №5 - VNS Lab 1 - Task 2-16**

**Задача:**

Обчислити значення виразів. Пояснити отримані результати.

1) ++n\*++m

2) m++<n

3) n++>m

*2) Дизайн та планована оцінка часу виконання завдань*

**Завдання №1 - Class Practice Work - Особистий порадник**

Дизайн алгоритму завдання прикріплений до папки “ Diagrams ” під назвою practice\_work\_team\_tasks\_vladyslav\_murashko.drawio

Попередньо розраховував, що це завдання займе 1 годину

**Завдання №2 - Self Practice Work – Algotester lab 2 variant 3**

Дизайн алгоритму завдання прикріплений до папки “Diagrams” під назвою self\_practice\_work\_algotester\_task\_1\_vladyslav\_murashko.drawio

Попередньо розраховував, що це завдання займе 30 хв

**Завдання №3 – Algotester lab 1 variant 3**

Дизайн алгоритму завдання прикріплений до папки “Diagrams” під назвою algotester\_lab\_1\_variant\_3\_vladyslav\_murashko.drawio

Попередньо розраховував, що це завдання займе 30 хв

**Завдання №4 - VNS Lab 1 - Task 1-16**

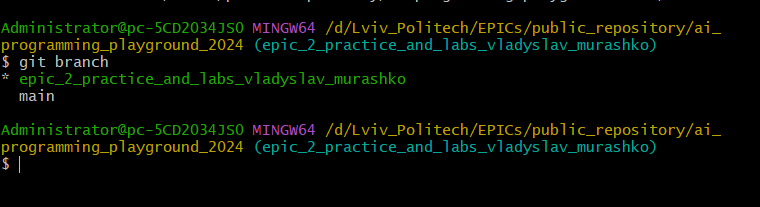
Дизайн алгоритму завдання прикріплений до папки “Diagrams” під назвою vns\_lab\_1\_task\_1\_variant\_16\_vladyslav\_murashko.drawio

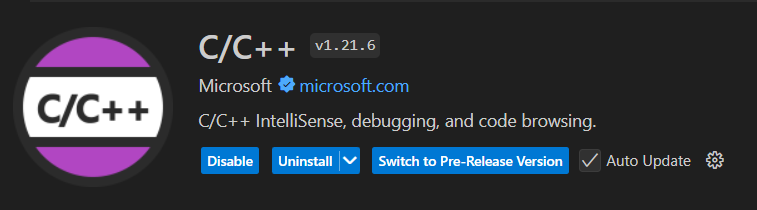
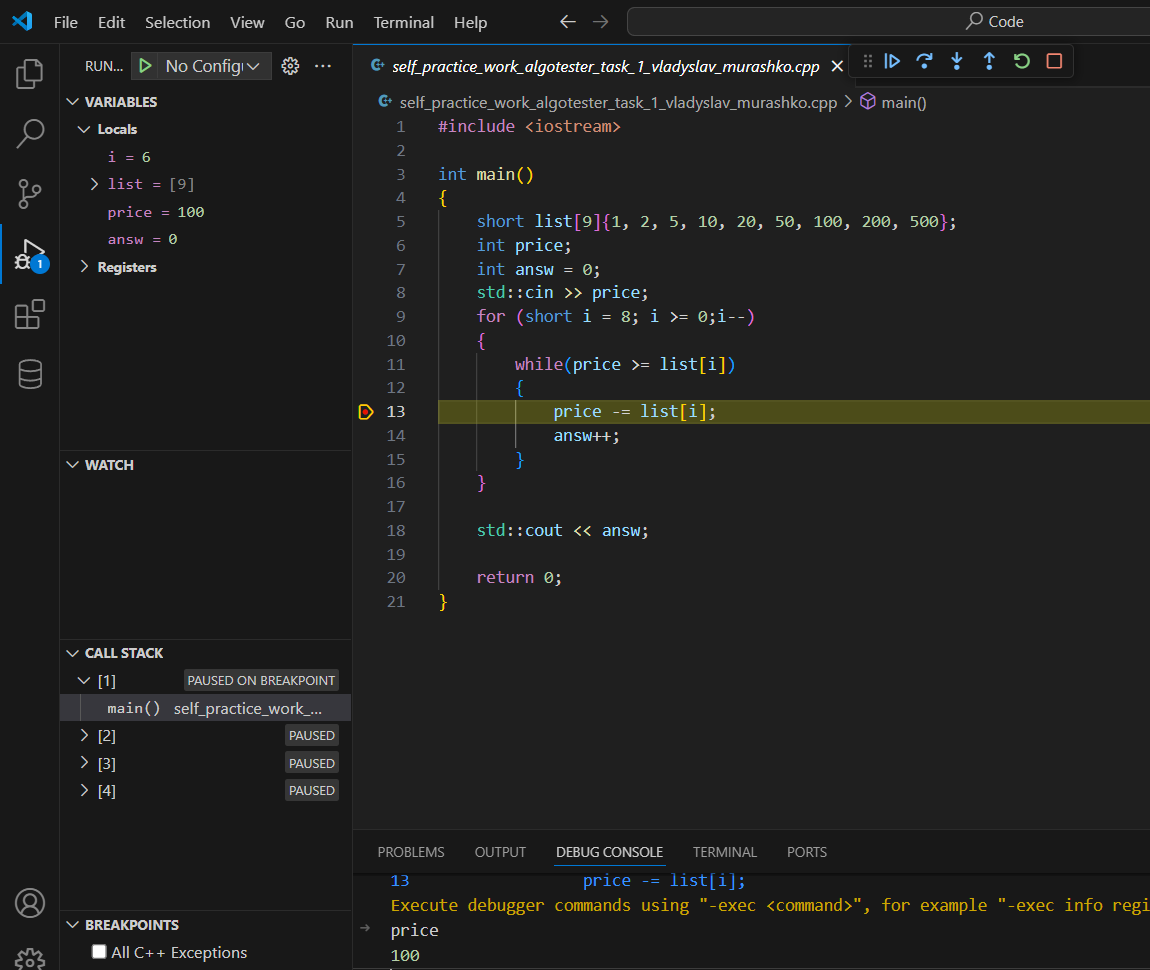
Попередньо розраховував, що це завдання займе 30 хв

**Завдання №5 - VNS Lab 1 - Task 2-16**

Дизайн алгоритму завдання прикріплений до папки “Diagrams” під назвою vns\_lab\_1\_task\_2\_variant\_16\_vladyslav\_murashko.drawio

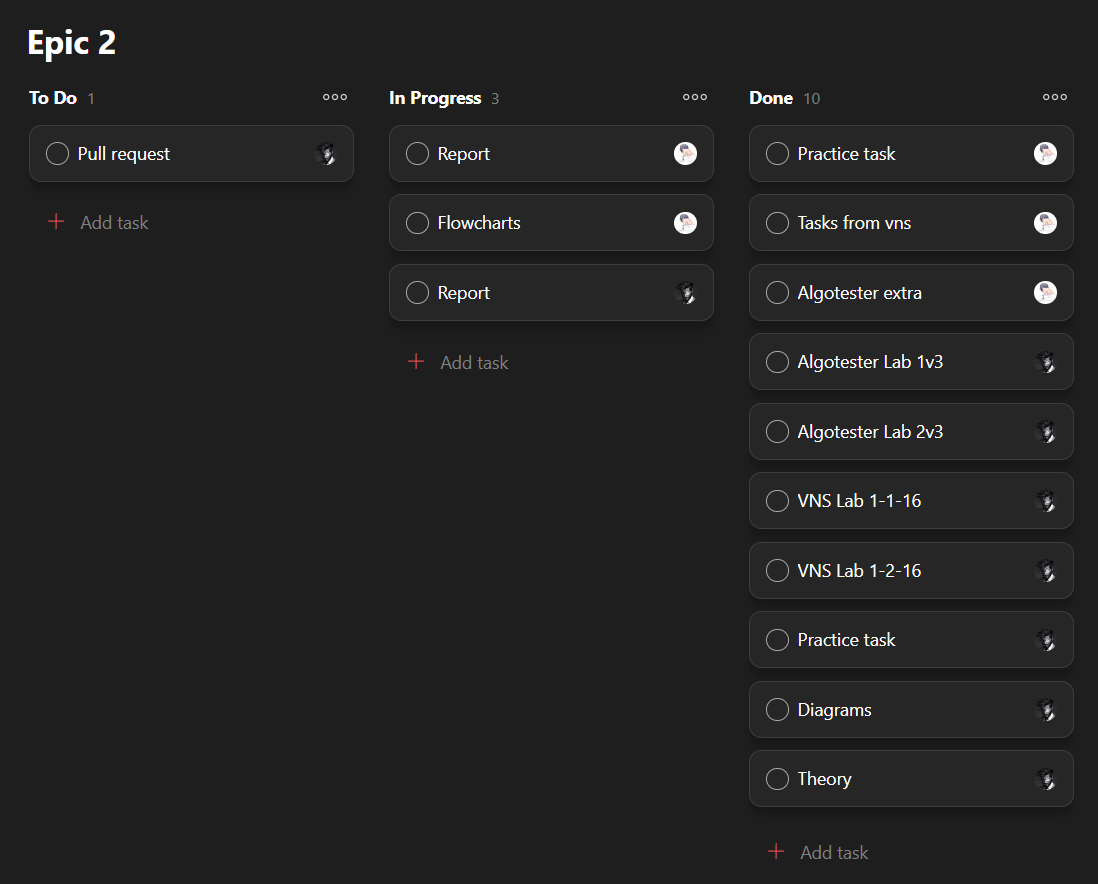
Попередньо розраховував, що це завдання займе 30 хв

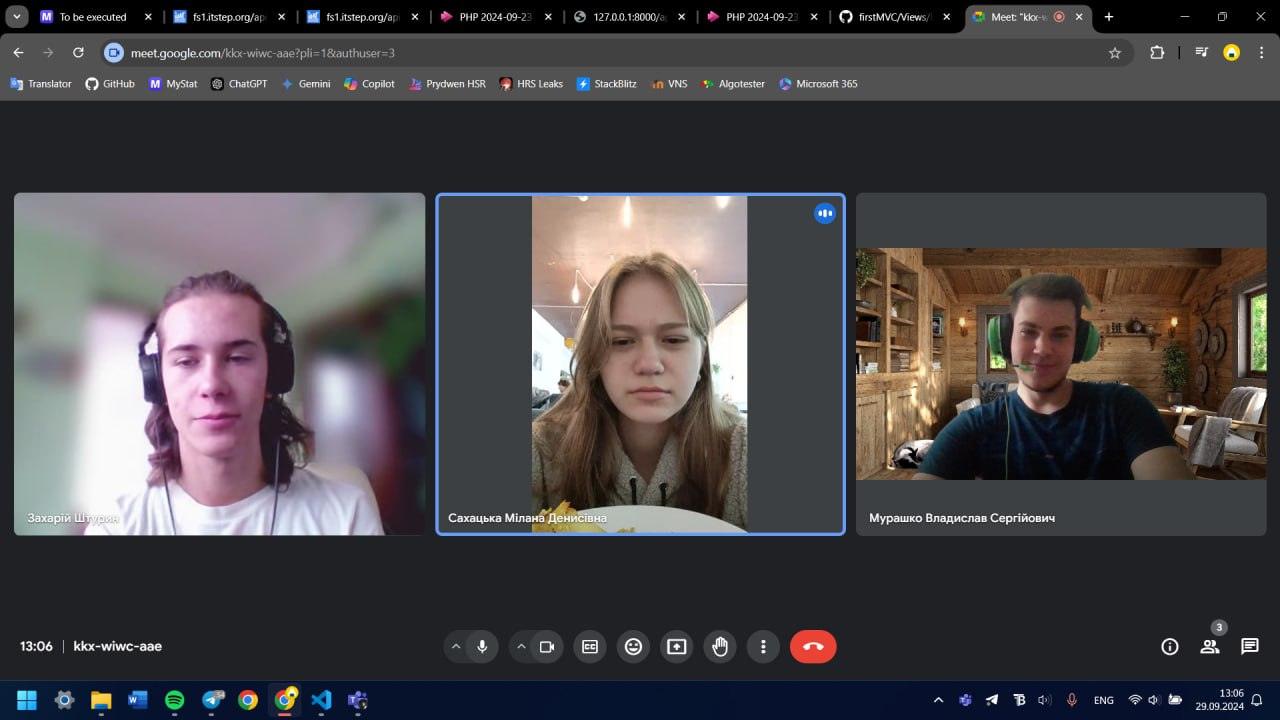
*3) Конфігурація середовища для виконання завдань*

Git та моя гілка

Налаштований дебагер

Встановлене розширення C/C++ для VS Code



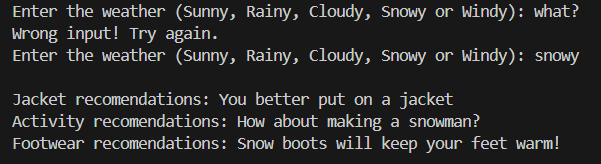
Командна дошка задач

Скрін зустрічі нашої команди

*4) Код програми з посиланням на зовнішні ресурси*

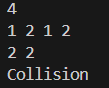
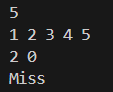
* **До завдання №1 код practice\_work\_task\_1\_vladyslav\_murashko.cpp**
* **До завдання №2 код self\_practice\_work\_algotester\_task\_1\_vladyslav\_murashko.cpp**
* **До завдання №3 код algotester\_lab\_1\_variant\_3\_vladyslav\_murashko.cpp**
* **До завдання №4 код vns\_lab\_1\_task\_1\_variant\_16\_vladyslav\_murashko.cpp**
* **До завдання №5 код vns\_lab\_1\_task\_2\_variant\_16\_vladyslav\_murashko.cpp**

*5) Результати виконаних завдань, тестування та фактично затрачений час*

** Завдання №1 - Class Practice Work - Особистий порадник**

Зайняло часу – 1 година

**Завдання №2 - Self Practice Work – Algotester lab 2 variant 3**

********

Зайняло часу – 30 хв.

**Завдання №3 – Algotester lab 1 variant 3**

****

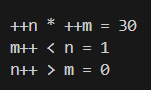
Зайняло часу – 30 хв.

**Завдання №4 – VNS Lab 1 - Task 1-16**

****

Зайняло часу – 30 хв.

**Завдання №5 – VNS Lab 1 - Task 2-16**

****

При

n = 5 та m = 4.

Зайняло часу – 30 хв.

**Висновок:** Я навчився писати лінійні та розгалужені алгоритми, використовувати умовні та логічні оператори, змінні та константи, навчився різниці між типами даних. Попрацював з вводом ти виводом даних, коментарями, виконанням базових операцій та використанням вбудованих функцій.