Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту  
  
A blue and white logo

Description automatically generated

**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:

ВНС Лабораторної Роботи № 1

Алготестер Лабораторної Роботи № 1

Практичних Робіт до блоку № 2

**Виконала:**

Студентка групи ШІ-11

Гуменюк Анастасія Олександрівна

**Тема роботи:** Ознайомлення з різними системами числення, такими як двійкова, вісімкова, десяткова та шістнадцяткова. Ознайомлення з етапами компіляції: препроцесор, компіляція. Використання бібліотек (stdio, cmath, iostream) та їх роль у програмуванні. Основи роботи з функціями cin, cout, printf, scanf та форматуванням виводу. Важливість, види та правила написання ефективних коментарів в коді. Структура, особливості, умовні оператори (if, else, switch-case) та логічні оператори (AND, OR, NOT).

**Мета роботи:** Ознайомитися з етапами компіляції, роботою препроцесора та директивами, а також зрозуміти функції компілятора для оптимізації процесу написання програм. Засвоїти принципи оголошення змінних і констант, вибору відповідного типу даних (int, char, float, double). Навчитися використовувати стандартні бібліотеки C++ (iostream, cmath, stdio) для полегшення програмування та виконання стандартних операцій. Опанувати роботу з операторами вводу/виводу (cin, cout, printf, scanf), а також форматуванням виводу даних у різних типах. Розуміти важливість коментарів у програмуванні та навчитися ефективно коментувати код для полегшення його підтримки. Опанувати лінійні алгоритми, їхню структуру та написання послідовних операцій у програмах C++. Засвоїти використання розгалужень та умовних операторів для побудови складніших алгоритмів. Вивчити логічні оператори та застосовувати їх у програмних умовах для управління логікою виконання алгоритмів.

**Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Використання систем числення в програмуванні.
* Тема № 2: Функції компілятора і етапи компіляції.
* Тема № 3: Змінні, Константи, Типи даних та їх розміри.
* Тема № 4:  Бібліотеки в C++.
* Тема № 5: Ввід та Вивід даних. Робота з різними типами даних у вводі/виводі.
* Тема № 6: Коментарі у Коді.  Види коментарів у С++.
* Тема № 7: Розгалужені алгоритми та Умовні Оператори.
* Тема № 8: Визначення та види логічних операторів.

1. Індивідуальний план опрацювання теорії:

Тема №1: Використання систем числення в програмуванні.

* + Джерела:

<https://ua5.org/osnprog/28-ponjattja-sistemi-chislennja.html>

<https://www.youtube.com/watch?v=yGmVLDenVpE>

* + - Що опрацьовано:
      * Позиційна і непозиційна системи числення.
      * Система числення, що використовується в комп’ютерах.
      * Вісімкова система числення в Linux (права доступу до файлів).
    - Статус: Ознайомлена
    - Початок опрацювання теми: 15.10.2024.
    - Звершення опрацювання теми: 15.10.2024 (25хв.).

Тема №2: Функції компілятора і етапи компіляції.

* + Джерела:

<https://coworkingclub.com.ua/uk/scho-take-kompilyator/>

<https://www.guru99.com/uk/compiler-design-phases-of-compiler.html>

* + - Що опрацьовано:
      * Компілятор та інтерпретатор – різниця.
      * Шість фаз компілятора це 1) лексичний аналіз 2) синтаксичний аналіз 3) семантичний аналіз 4) проміжний генератор коду 5) оптимізатор коду 6) код Generator.
    - Статус: Ознайомлена
    - Початок опрацювання теми: 15.10.2024.
    - Звершення опрацювання теми: 15.10.2024 (30хв.).

Тема №3: Змінні, Константи, Типи даних та їх розміри.

* + Джерела:

<https://www.guru99.com/uk/c-variable-datatypes-constants.html>

<https://acode.com.ua/urok-36-typy-danyh-z-plavayuchoyu-krapkoyu-float-double-i-long-double/>

* + - Що опрацьовано:
      * Чотири типи даних, які зазвичай використовуються
      * Типи констант (перед вісімковою 0, перед шістнадцятковою 0X, символьні в одиничних лапках, рядкові в подвійних лапках).
      * Оголошення констант.
      * Float – 6-9 значень точності(частіше 7), double – 15-18 цифр(частіше 16), long double: 15, 18 чи 33.
    - Статус: Ознайомлена
    - Початок опрацювання теми: 17.10.2024.
    - Звершення опрацювання теми: 17.10.2024 (30хв.).

Тема №4: Бібліотеки в C++.

* + Джерела:

<https://www.youtube.com/watch?v=m-WJikuZGuU>

* + - Що опрацьовано:
      * Стандартна бібліотека C — кожний заголовок стандартної бібліотеки C включено у стандартну бібліотеку C++ під іншою назвою: вилучено закінчення .h і додано літеру 'c' на початку.
      * Наприклад, Сmath та math.h, якщо написати math.h, то використати змінну pow вже не можна, адже існує така функція, з Сmath таких проблем немає. Усі функції з cmath знаходяться у просторі імен std. Тому, щоб використовувати функцію pow, потрібно вказати, що це std::pow. Тому, можна використовувати ім'я pow як змінну в іншому контексті без конфліктів, оскільки компілятор знає, що є різниця між глобальною змінною і функцією з простору імен std.
    - Статус: Ознайомлена
    - Початок опрацювання теми: 17.10.2024.
    - Звершення опрацювання теми: 17.10.2024 (35хв.).

Тема №5: Ввід та Вивід даних. Робота з різними типами даних у вводі/виводі.

* + Джерела:

<https://acode.com.ua/urok-215-potoky-vvodu-i-vyvodu/>

<https://w3schoolsua.github.io/c/c_data_types.html#gsc.tab=0>

<https://studfile.net/preview/5775175/page:10/>

<https://www.freecodecamp.org/ukrainian/news/spetsyfikatory-formatu-v-c/>

* + - Що опрацьовано:
      * Класи istream(Оператор вилучення >> значень з потоку), ostream (Оператор вставки <<  значень в потік) , iostream.
      * Потік вводу та виводу.
      * Специфікатори типів даних (%c - одиничний символ, %s -рядок, %d та %і десяткове ціле число, %p - адреса (або вказівник), %f - число з рухомою комою і тд).
      * Scanf та printf структура. Scanf("%d %f", &a, &b), printf("a: %d, b: %.2f\n", a, b);
    - Статус: Ознайомлена
    - Початок опрацювання теми: 15.10.2024.
    - Звершення опрацювання теми: 15.10.2024 (20хв.).

Тема №6: Коментарі у Коді.  Види коментарів у С++.

* + Джерела:

<https://w3schoolsua.github.io/c/c_comments.html#google_vignette>

* + - Що опрацьовано:
      * Однорядкові коментарі починаються з двох скісних рисок (//).
      * Багаторядкові коментарі починаються з /\* і закінчуються \*/
    - Статус: Ознайомлена
    - Початок опрацювання теми: 20.10.2024.
    - Звершення опрацювання теми: 20.10.2024 (5хв.).

Тема №7: Розгалужені алгоритми та Умовні Оператори.

* + Джерела:

<https://acode.com.ua/urok-67-operatory-umovnogo-rozgaluzhennya-if-else/>

<https://acode.com.ua/urok-68-operator-switch/>

<https://acode.com.ua/urok-44-umovnyj-ternarnyj-operator-sizeof-i-koma/>

* + - Що опрацьовано:
      * Умовні розгалуження if/else.
      * Оператор switch, лейбли case.
      * switch і fall-through, **switch і оператор break**
      * Тарнарний оператор.
    - Статус: Ознайомлена
    - Початок опрацювання теми: 20.10.2024.
    - Звершення опрацювання теми: 20.10.2024 (25хв.).
* Тема №8: Визначення та види логічних операторів.
  + Джерела:

<https://acode.com.ua/urok-46-logichni-operatory-i-abo-ne/>

* + - Що опрацьовано:
      * Використання операторів І/АБО/НЕ
    - Статус: Ознайомлена
    - Початок опрацювання теми: 20.10.2024.
    - Звершення опрацювання теми: 20.10.2024 (15хв.).

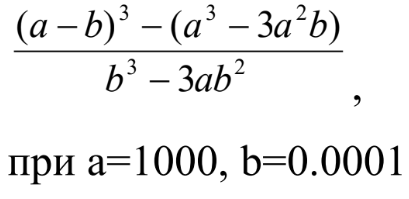
**Виконання роботи:**

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

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

**Задача:**

Обчислити значення виразів. Пояснити отримані результати.

1) ++n\*++m

2) m++<n

3) n++>m

**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 - 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 №5 - Self Practice Work – Algotester Щасливий результат**

**Задача:**

Після завершення основного туру олімпіади з програмування Зеник отримав невеличкий клаптик паперу, на якому було надруковане число xx — кількість балів, що набрав Зеник. Зауважте, що згідно з кращими традиціями олімпіади з програмування, кількість балів Зеника не може бути нульовою чи від’ємною.

Помітивши не дуже щасливе обличчя Зеника, Марічка нагадала йому про щасливі цифри. Як ви вже напевно знаєте, щасливими вважають цифри 4 та 7. Марічка запевнила Зеника, що найкращим є не найбільший результат, а той, десятковий запис якого містить найбільше щасливих цифр.

Вам необхідно допомогти юному учаснику олімпіади з програмування та порахувати кількість щасливих цифр у його результаті.

**Вхідні дані**

У єдиному рядку задано одне ціле число xx — результат Зеника.

**Вихідні дані**

У єдиному рядку виведіть одне ціле число — кількість щасливих цифр у десятковому записі xx.

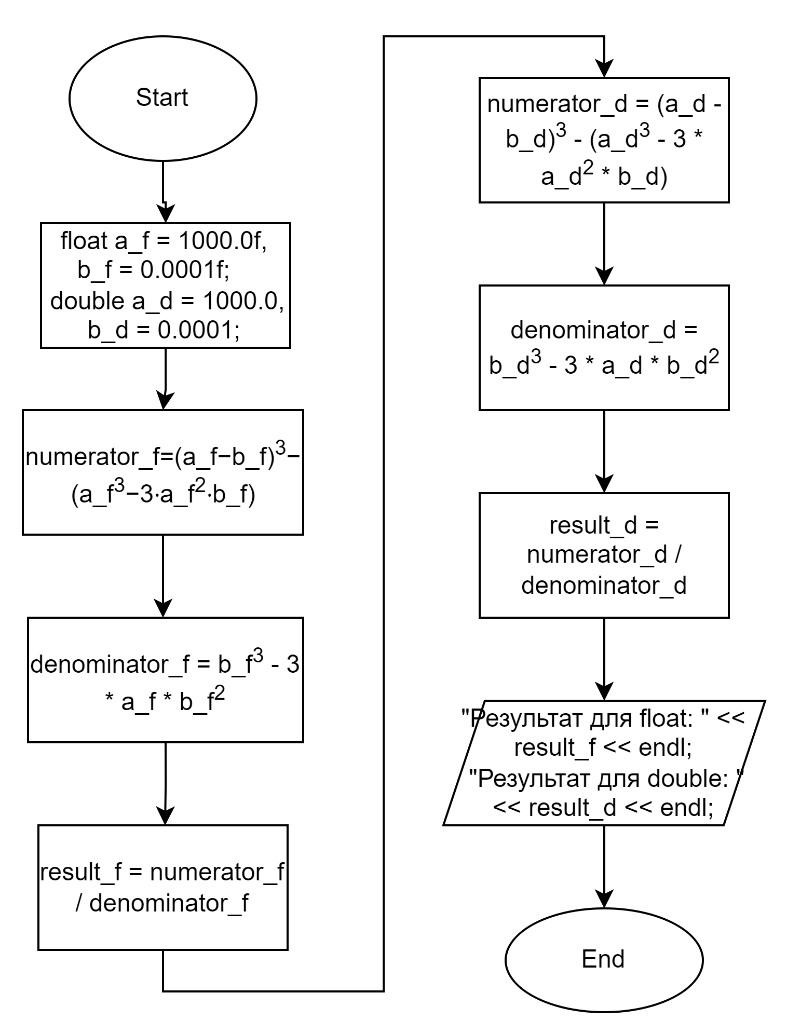
**Обмеження** 1≤x≤109.

1. *Дизайн та планована оцінка часу виконання завдань:*

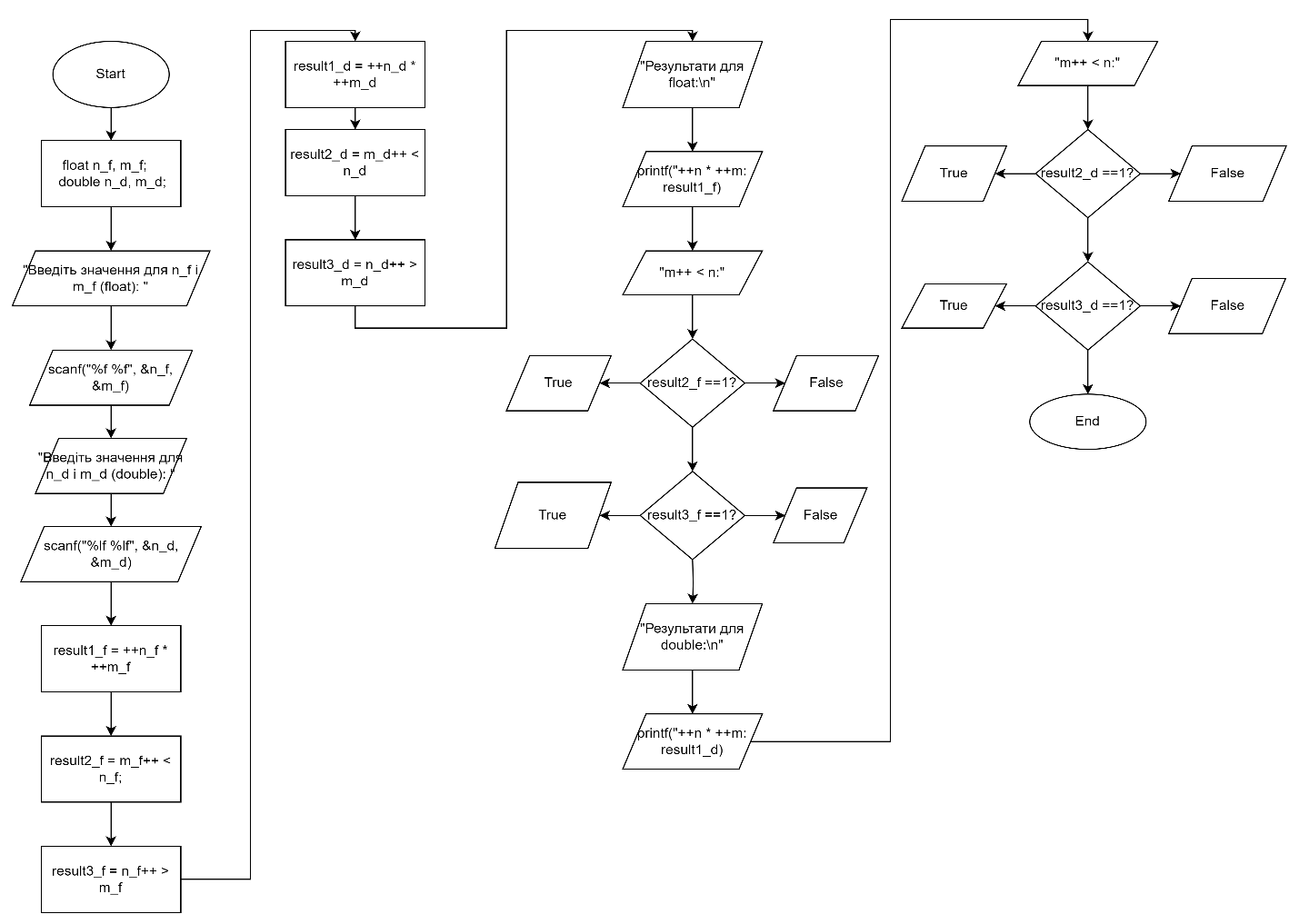
**Дошка в Trello:**

****

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

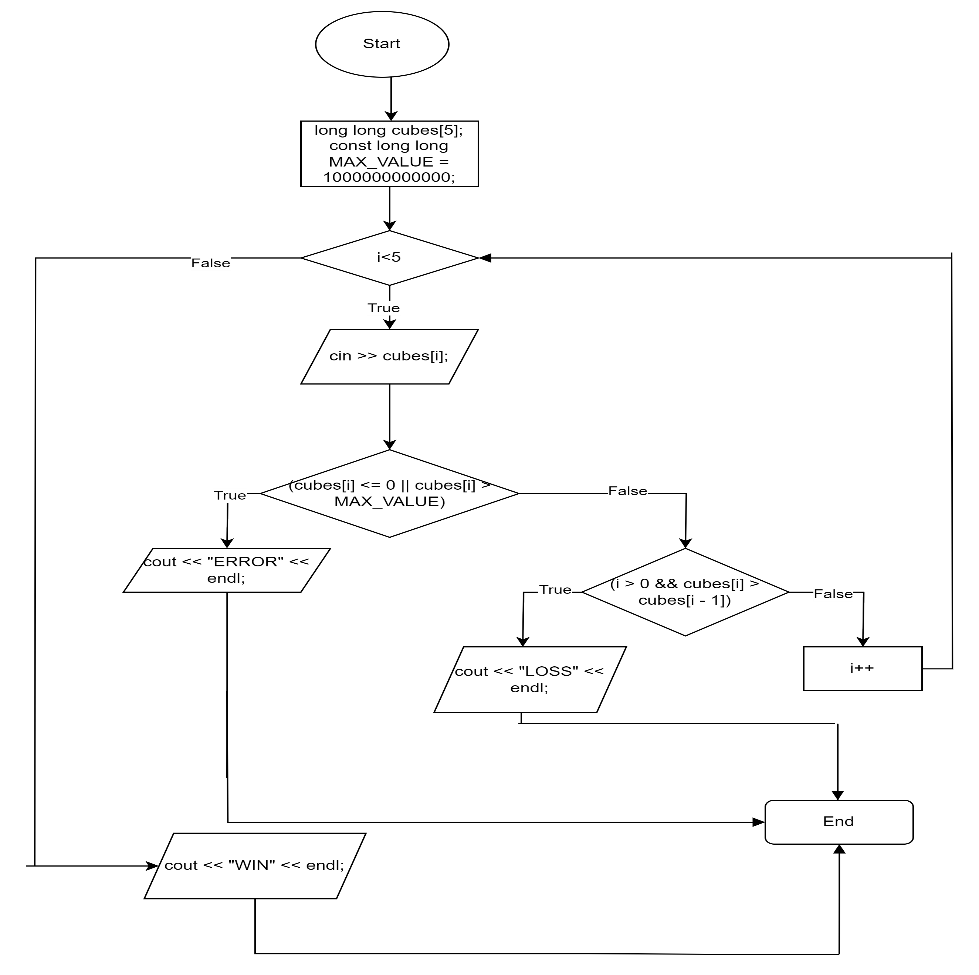
**** Планований час: 30 хв.

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

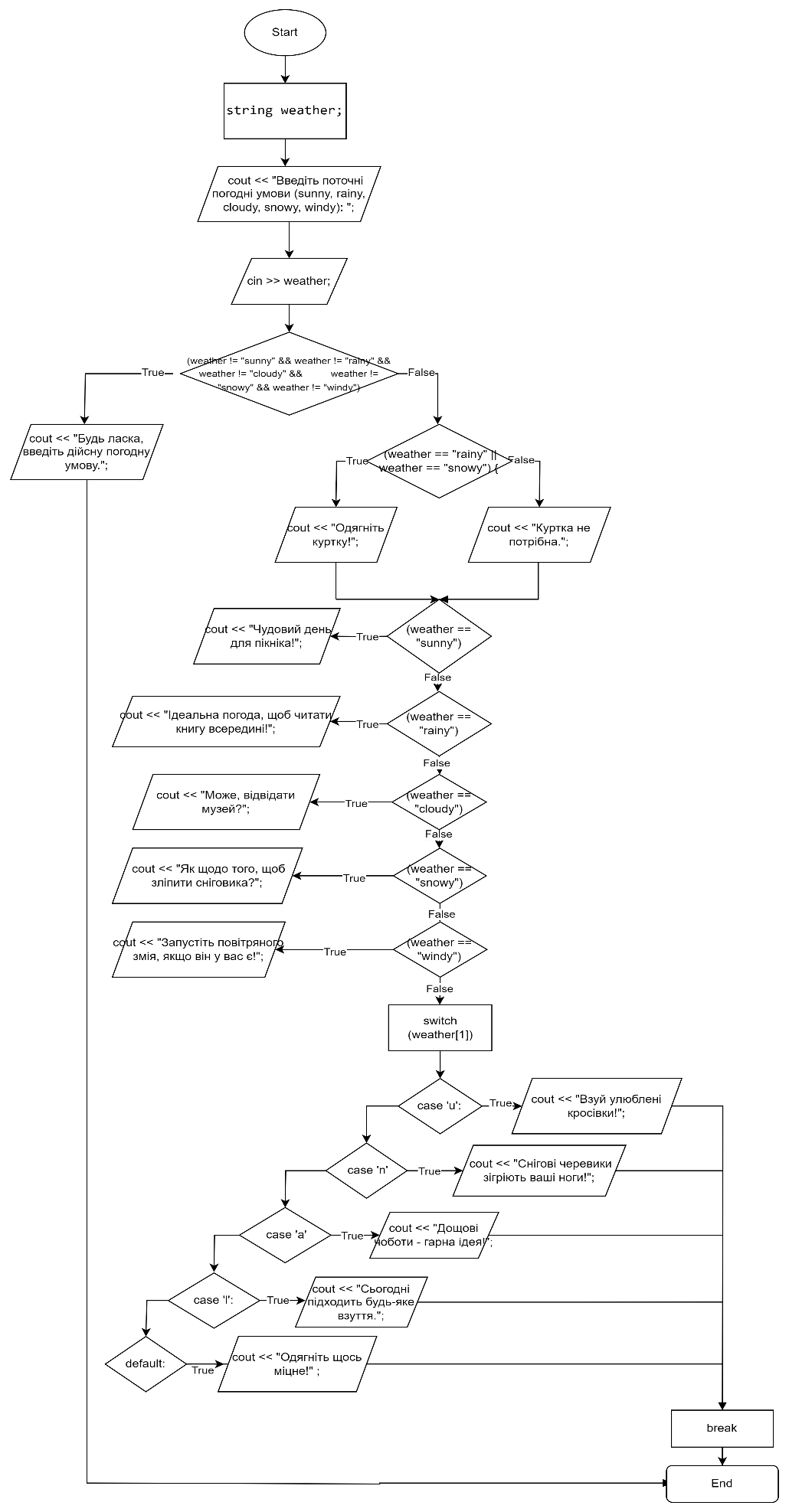


Планований час: 30-35 хв.

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

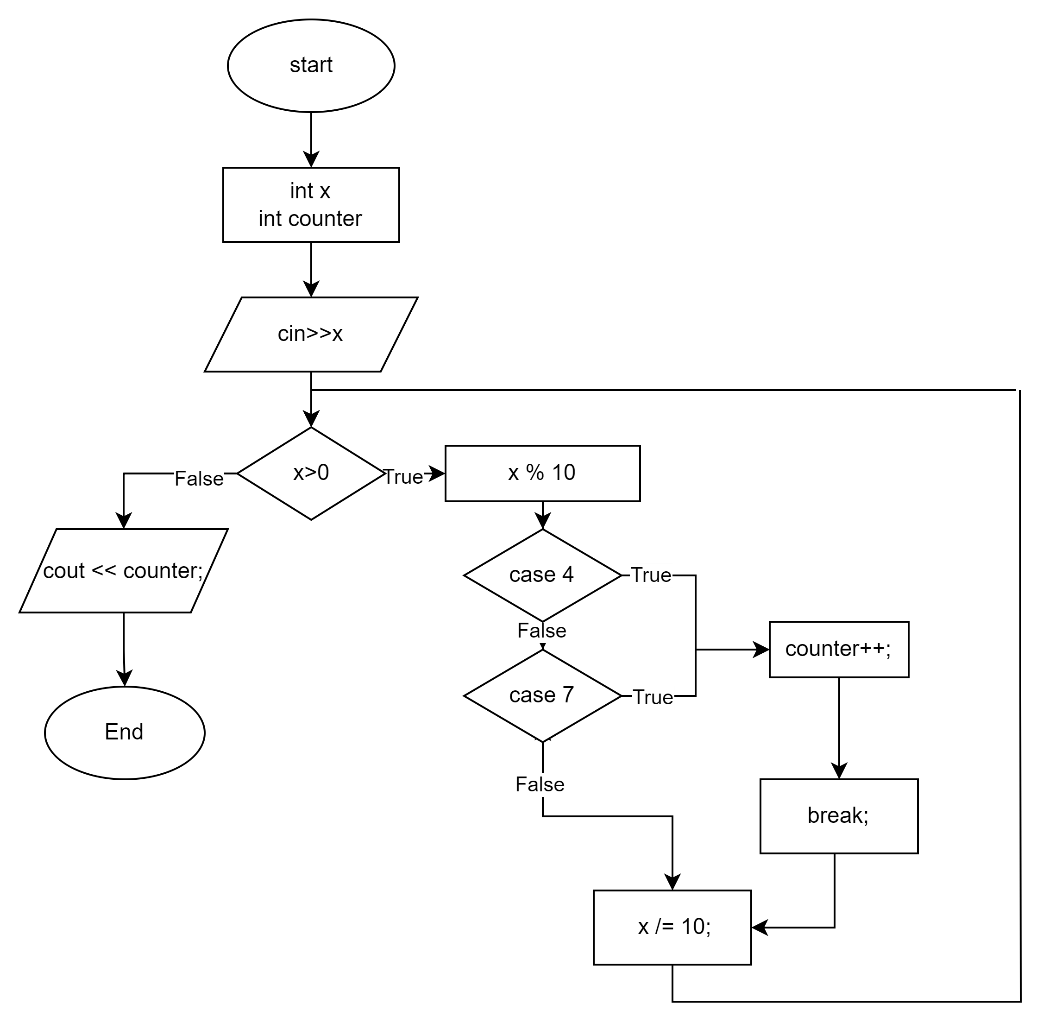
Планований час: 20 хв.

**Завдання №4 - Class Practice Work - Особистий порадник**

Планований час: 20 хв.

**Примітка:** switch використовує другу літеру (індекс 1) погодних умов для визначення типу взуття.

**Завдання №5 - Self Practice Work – Algotester Щасливий результат**



Планований час:10 хв.

1. *Код програм з посиланням на зовнішні ресурси*

**Завдання №1**

#include <iostream>

#include <cmath>

using namespace std;

int main() {

    float a\_f = 1000.0f, b\_f = 0.0001f;

    double a\_d = 1000.0, b\_d = 0.0001;

    //float

    float numerator\_f = pow(a\_f - b\_f, 3) - (pow(a\_f, 3) - 3 \* pow(a\_f, 2) \* b\_f);

    float denominator\_f = pow(b\_f, 3) - 3 \* a\_f \* pow(b\_f, 2);

    float result\_f = numerator\_f / denominator\_f;

    //double

    double numerator\_d = pow(a\_d - b\_d, 3) - (pow(a\_d, 3) - 3 \* pow(a\_d, 2) \* b\_d);

    double denominator\_d = pow(b\_d, 3) - 3 \* a\_d \* pow(b\_d, 2);

    double result\_d = numerator\_d / denominator\_d;

    cout << "Результат для float: " << result\_f << endl;

    cout << "Результат для double: " << result\_d << endl;

    return 0;

}

Тип даних float має меншу точність у порівнянні з double (близько 7 значущих цифр для float і 15 значущих цифр для double).

<https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/epic_2_practice_and_labs_anastasiia_humeniuk/ai_11/anastasiia_humeniuk/epic2/vns_lab_1_task_1_variant_16_anastasiia_humeniuk.cpp>

**Завдання №2**

#include <iostream>

#include <cstdio>

using namespace std;

int main() {

    float n\_f, m\_f;

    double n\_d, m\_d;

    printf("Введіть значення для n\_f і m\_f (float): ");

    scanf("%f %f", &n\_f, &m\_f);

    printf("Введіть значення для n\_d і m\_d (double): ");

    scanf("%lf %lf", &n\_d, &m\_d);

    // Обчислення для float

    float result1\_f = ++n\_f \* ++m\_f;

    bool result2\_f = m\_f++ < n\_f;

    bool result3\_f = n\_f++ > m\_f;

    // Обчислення для double

    double result1\_d = ++n\_d \* ++m\_d;

    bool result2\_d = m\_d++ < n\_d;

    bool result3\_d = n\_d++ > m\_d;

    printf("Результати для float:\n");

    printf("++n \* ++m: %.2f\n", result1\_f);

    printf("m++ < n: %s\n", result2\_f ? "true" : "false");

    printf("n++ > m: %s\n", result3\_f ? "true" : "false");

    printf("Результати для double:\n");

    printf("++n \* ++m: %.2lf\n", result1\_d);

    printf("m++ < n: %s\n", result2\_d ? "true" : "false");

    printf("n++ > m: %s\n", result3\_d ? "true" : "false");

    return 0;

}

<https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/epic_2_practice_and_labs_anastasiia_humeniuk/ai_11/anastasiia_humeniuk/epic2/vns_lab_1_task_2_variant_16_anastasiia_humeniuk.cpp>

**Завдання №3**

#include <iostream>

using namespace std;

int main() {

    long long cubes[5];

    const long long MAX\_VALUE = 1000000000000;

    for (int i = 0; i < 5; ++i) {

        cin >> cubes[i];

        if (cubes[i] <= 0 || cubes[i] > MAX\_VALUE) {

            cout << "ERROR" << endl;

            return 0;

        }

        if (i > 0 && cubes[i] > cubes[i - 1]) {

            cout << "LOSS" << endl;

            return 0;

        }

    }

    cout << "WIN" << endl;

    return 0;

}

<https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/epic_2_practice_and_labs_anastasiia_humeniuk/ai_11/anastasiia_humeniuk/epic2/algotester_lab_1_variant_3_anastasiia_humeniuk.cpp>

**Завдання №4**

#include <iostream>

#include <string>

using namespace std;

int main() {

    string weather;

    cout << "Введіть поточні погодні умови (sunny, rainy, cloudy, snowy, windy): ";

    cin >> weather;

      if (weather != "sunny" && weather != "rainy" && weather != "cloudy" &&

        weather != "snowy" && weather != "windy") {

        cout << "Будь ласка, введіть дійсну погодну умову." << endl;

        return 1;

    }

    if (weather == "rainy" || weather == "snowy") {

        cout << "Одягніть куртку!" << endl;

    } else {

        cout << "Куртка не потрібна." << endl;

    }

    if (weather == "sunny") {

        cout << "Чудовий день для пікніка!" << endl;

    } else if (weather == "rainy") {

        cout << "Ідеальна погода, щоб читати книгу всередині!" << endl;

    } else if (weather == "cloudy") {

        cout << "Може, відвідати музей?" << endl;

    } else if (weather == "snowy") {

        cout << "Як щодо того, щоб зліпити сніговика?" << endl;

    } else if (weather == "windy") {

        cout << "Запустіть повітряного змія, якщо він у вас є!" << endl;

    }

switch (weather[1]) {

        case 'u':

            cout << "Взуй улюблені кросівки!" << endl;

            break;

        case 'n':

             cout << "Снігові черевики зігріють ваші ноги!" << endl;

            break;

        case 'a':

            cout << "Дощові чоботи - гарна ідея!" << endl;

            break;

        case 'l':

            cout << "Сьогодні підходить будь-яке взуття." << endl;

            break;

        default:

            cout << "Одягніть щось міцне!" << endl;

            break;

    }

    return 0;

}

<https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/epic_2_practice_and_labs_anastasiia_humeniuk/ai_11/anastasiia_humeniuk/epic2/practice_work_task_1_anastasiia_humeniuk.cpp>

**Завдання №5**

#include <iostream>

using namespace std;

int main() {

    int x;

    int counter = 0;

    cin >> x;

    while (x > 0) {

        switch (x % 10) {

            case 4:

            case 7:

                counter++;

                break;

        }

        x /= 10;

    }

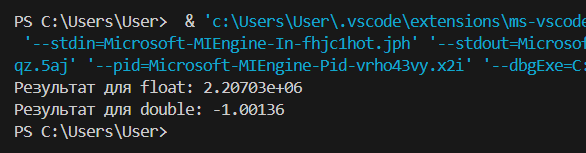
    cout << counter;

}

<https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/epic_2_practice_and_labs_anastasiia_humeniuk/ai_11/anastasiia_humeniuk/epic2/self_practice_work_algotester_task_1_anastasiia_humeniuk.cpp>

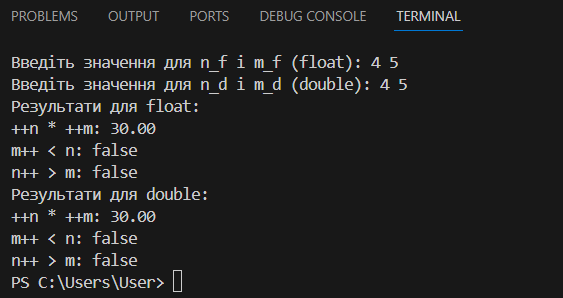
1. *Результати виконання завдань, тестування та фактично затрачений час:*

**Завдання №1  VNS Lab 1 Task 1**

****

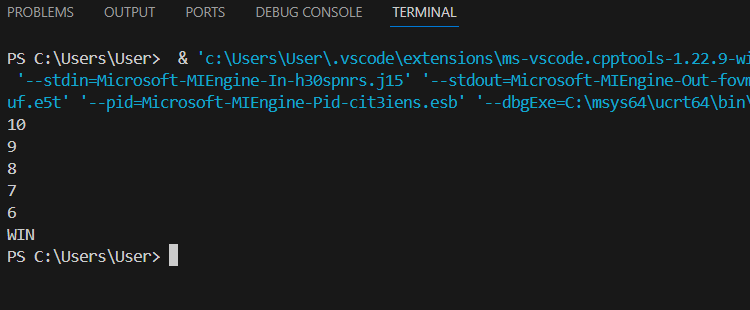
Фактично затрачений час: 15 хв.

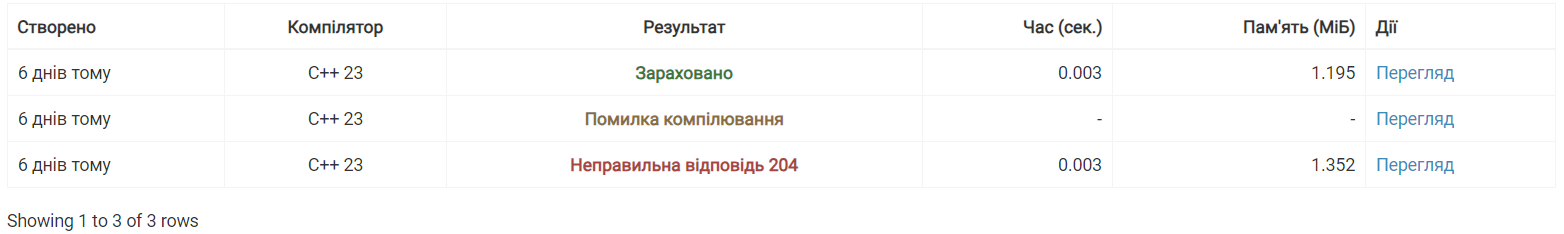
**Завдання №2 VNS Lab 1 Task 2**

**

Фактично затрачений час: 25 хв.

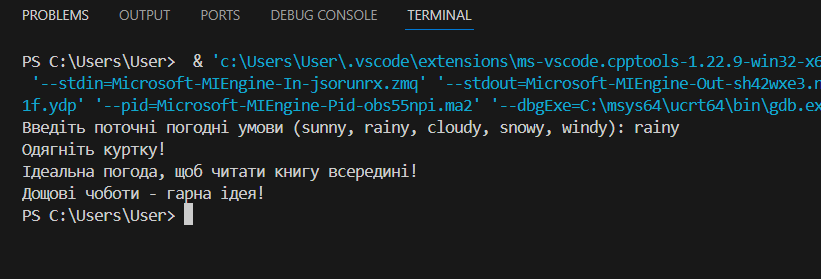
**Завдання №3 Algotester Lab 1 Task 1**

**

**

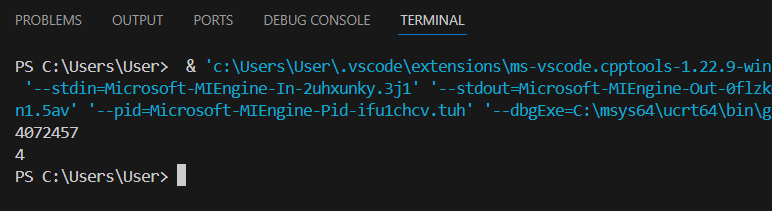
Фактично затрачений час: 20 хв.

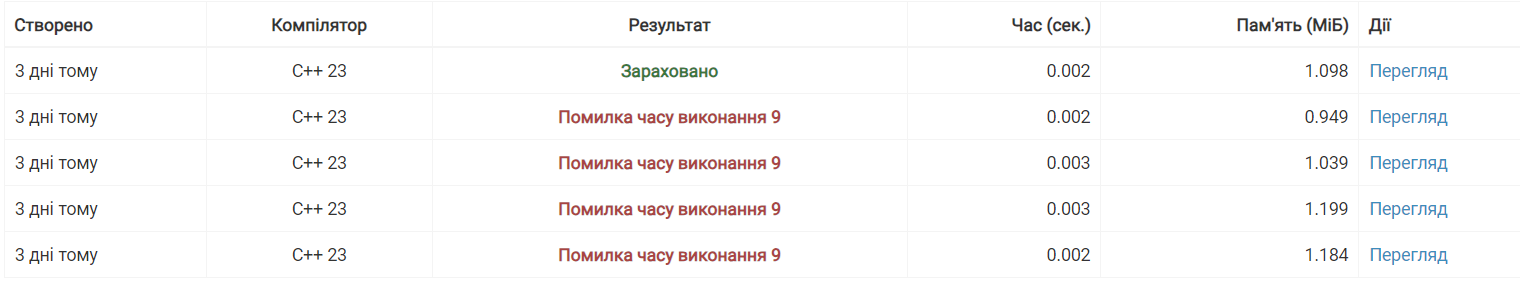
**Завдання №4 Class Practice Task**

**

Фактично затрачений час: 30 хв.

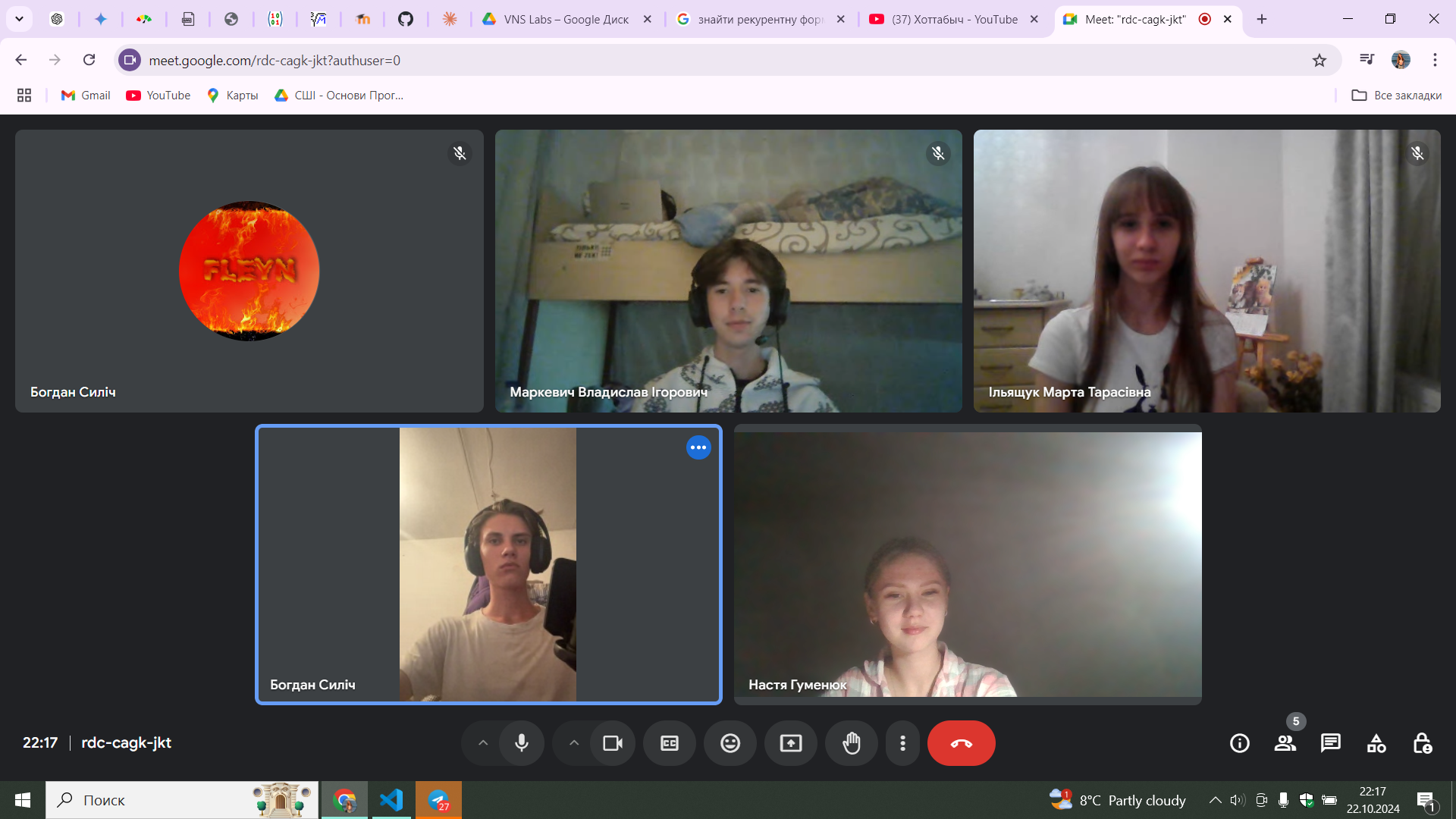
* **Завдання №5 Self Practice Task**

**

**

Фактично затрачений час: 35 хв.

1. *Кооперація з командою:*



**Висновок:** Під час виконання другого епіку я ознайомилася з основними етапами компіляції та роботою препроцесора, що дало мені краще розуміння того, як працює компілятор та які функції він виконує для оптимізації коду. Я засвоїла принципи оголошення змінних і констант, а також вибір відповідних типів даних, що є важливими для ефективної роботи з пам'яттю. Зрозуміла мету використання стандартних бібліотек, таких як iostream та cmath. Я оволоділа навичками форматування виводу, написання лінійних алгоритмів і використання умовних операторів для побудови складніших програм. А також на практиці відпрацювала структуру розгалуження, написавши декілька програм з їх використанням.