Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

про виконання

**Лабораторних та практичних робіт № 1**

***з дисципліни:*** «Мови та парадигми програмування»

***з розділу***: «Лінійні та розгалужені алгоритми. Умовні оператори. Константи, змінні»

***Виконав:***

Студент групи ШІ-13

Золотарчук Максим Володимирович

**Тема роботи:**

Ознайомитися з основними поняттями С++. Використати основні оператори мови С++. Використати масиви, роботу з рядками, створення та використання структур. Використати різні види умов та циклів для розв'язання завдань.

# **Мета роботи:**

Встановлення та налаштування середовища програмування (VS code) підключення MSYS2, ознайомлення з Git, GitHub, Algotester, Draw.io, Trello та реєстрація в них. Налагодження, створення й виконання простої програми, що містить базові оператори вводу й виводу та обчислення.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* **Тема №1**: VNS Lab 1 Task 1
* **Тема №2**: Algotester Lab 1 Task 2
* **Тема №3**:  Class Practice Task
* **Тема №4**: Self Practice Task

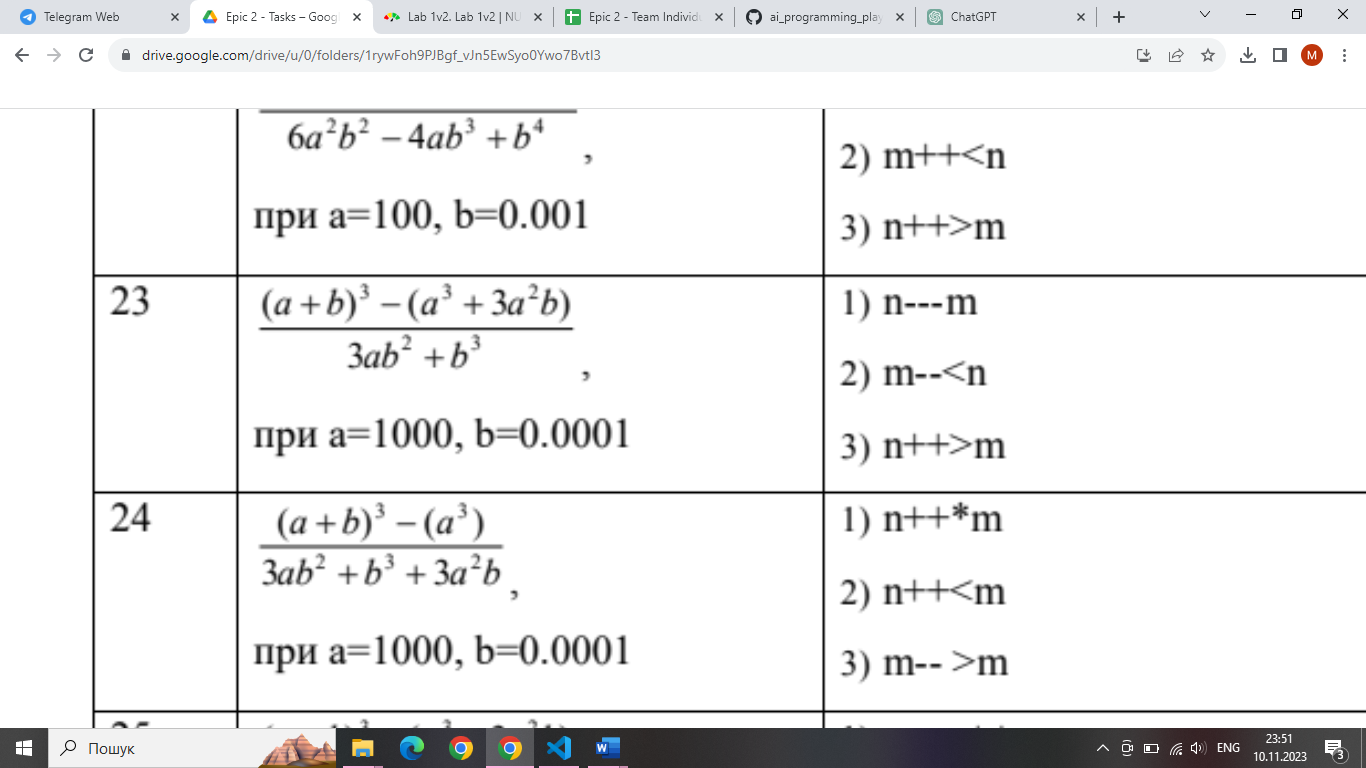
1. Індивідуальний план опрацювання теорії:

* **Тема №1:** VNS Lab 1 Task 1
* Джерела Інформації:
* Лекції
* Що опрацьовано: Обчислено значення виразу при різних дійсних типах даних (float і double).
* Статус: Ознайомлений.
* Початок опрацювання теми: 16.10.2023
* Звершення опрацювання теми: 27.10.2023
* **Тема №2:** VNS Lab 1 Task 2**.**
* Джерела Інформації:
* Практичне заняття
* Що опрацьовано:
* Обчислено значення виразів 1) n---m 2) m--<n 3) n++>m.
* Статус: Ознайомлений.
* Початок опрацювання теми: 19.10.2023
* Звершення опрацювання теми: 22.10.2023
* **Тема № 3:** Algotester Lab 1 Task 2.
* Джерела Інформації:
* Практичні
* Що опрацьовано: Виведено YES - якщо стіл буде стояти паралельно площині підлоги та довжина найменшої ніжки не буде рівна нулю. ERROR - у випадку якщо ви відпиляєте більшу довжину ніж має ніжка. NO - у інших випадках.
* Початок опрацювання теми: 18.10.2023
* Звершення опрацювання теми: 05.11.2023
* **Тема № 4:** Class Practice Task.
* Джерела Інформації:
  + - Практична
    - https://acode.com.ua/urok-67-operatory-umovnogo-rozgaluzhennya-if-else/
* Що опрацьовано: Створено простий порадник щодо погоди. Користувач вводить поточні погодні умови, а програма видає рекомендації щодо активності на основі погоди.
* Статус: Ознайомлений .
* Початок опрацювання теми: 20.10.2023
* Звершення опрацювання теми: 29.10.2023
* **Тема №5:** Self Practice Task.
* Джерела Інформації:Лекції
* Що опрацьовано:
* Статус: Ознайомлений.
* Початок опрацювання теми: Дата: 01.11.2023
* Звершення опрацювання теми: Дата: 07.11.2023

# **Виконання роботи:**

## **1. Опрацювання завдання та вимог до програм та середовища:**

Завдання №1 VNS Lab 1 Task 1

* Варіант завдання - 23
* Деталі завдання - Обчислити значення виразів. Пояснити отримані результати.
* 
* Різниця між float та double.

Завдання №2 Algotester Lab 1 Task 2.

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

Ви хочете зробити ніжки рівної довжини, для цього ви відпиляєте d� 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∗hminℎ���>=2∗ℎ��� 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.

* Завдання №3 Practice# programming: Class Practice Task
* Варіант завдання
* Деталі завдання:
* Epic 2 Practice - Оператори галуження
* Ви створюєте простий порадник щодо погоди. Користувач вводить поточні погодні умови, а програма видає рекомендації щодо активності на основі погоди.
* Вимоги
* Запропонувати користувачеві ввести поточні погодні умови.
* Варіанти
* ⁃ sunny;
* ⁃ rainy;
* ⁃ cloudy;
* ⁃ snowy;
* ⁃ windy;
* 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.
* Деталі логіки
* Рішення чи брати куртку (використовуючи if else)
* Якщо йде сніг або дощ, користувач повинен одягнути куртку.
* В іншому випадку куртка не потрібна.
* Рекомендація щодо активності (використання if, else if)
* Якщо сонячно, порекомендуйте «Чудовий день для пікніка!».
* Інакше, якщо буде дощ, рекомендуємо «Ідеальна погода, щоб читати книгу всередині!».
* Інакше, якщо хмарно, рекомендуємо «Може, відвідати музей?».
* Інакше, якщо сніг, порекомендуйте «Як щодо того, щоб зліпити сніговика?».
* Інакше, якщо буде вітер, порекомендуйте «Запустіть повітряного змія, якщо він у вас є!».
* Рекомендації щодо взуття (з використанням футляра для вимикача)
* sunny -> "Взуй улюблені кросівки!"
* rainy -> "Дощові чоботи - гарна ідея!"
* cloudy -> "Сьогодні підходить будь-яке взуття."
* snowy -> "Снігові черевики зігріють ваші ноги!"
* windy -> "Одягніть щось міцне!"
* Важливі деталі для врахування в імплементації програми

Завдання №5 Self-practice task: Зеник і Марічка вирішили зіграти n� разів у відому гру «камінь-ножиці-папір». Гра відбувається в декілька раундів. У кожному раунді кожен з учасників одночасно обирає один із трьох предметів: камінь, ножиці або папір. Результат раунду ґрунтується на предметах, обраних суперниками: камінь перемагає ножиці, ножиці перемагають папір, а папір перемагає камінь. Звернуть увагу, що в деякому раунді гравці можуть обрати той самий предмет. У такому випадку результатом раунду є нічия.

Марічка знає, у скількох раундах вона обере камінь, ножиці й папір, таку ж інформацію вона має про Зеника.

Марічка хоче дізнатися, скільки максимально раундів вона може виграти. Допоможіть їй дізнатися це.

## **2. Дизайн та планована оцінка часу виконання завдань:**

**Програма № 1**: VNS Lab 1 Task 1.

Блок-схема:

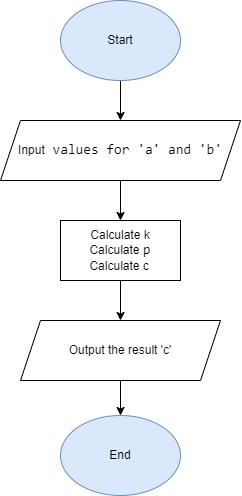


Figure Lab# programming: VNS Lab 1 Task 1.

* Планований час на реалізацію: 10 хвилин.

Figure VNS Lab 1 Task 2.

**Програма № 2**: Algotester Lab 1 Task 2.

Блок-схема:

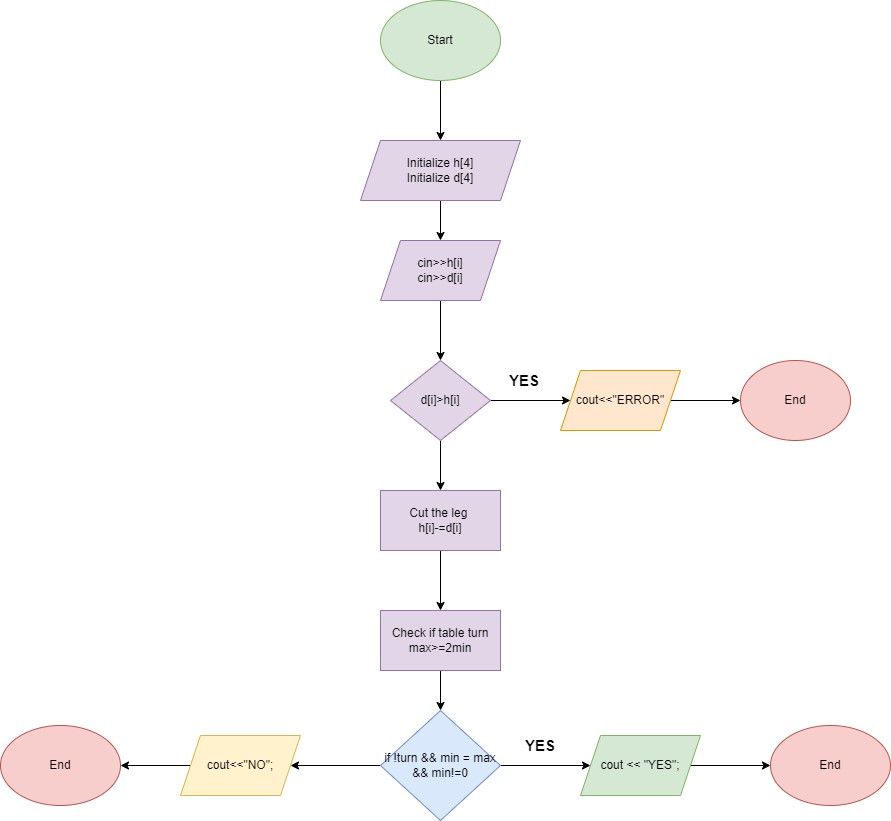


Рисунок Algotester Lab1

**Програма № 3**: Practice# programming: Class Practice Task.

Блок-схема:

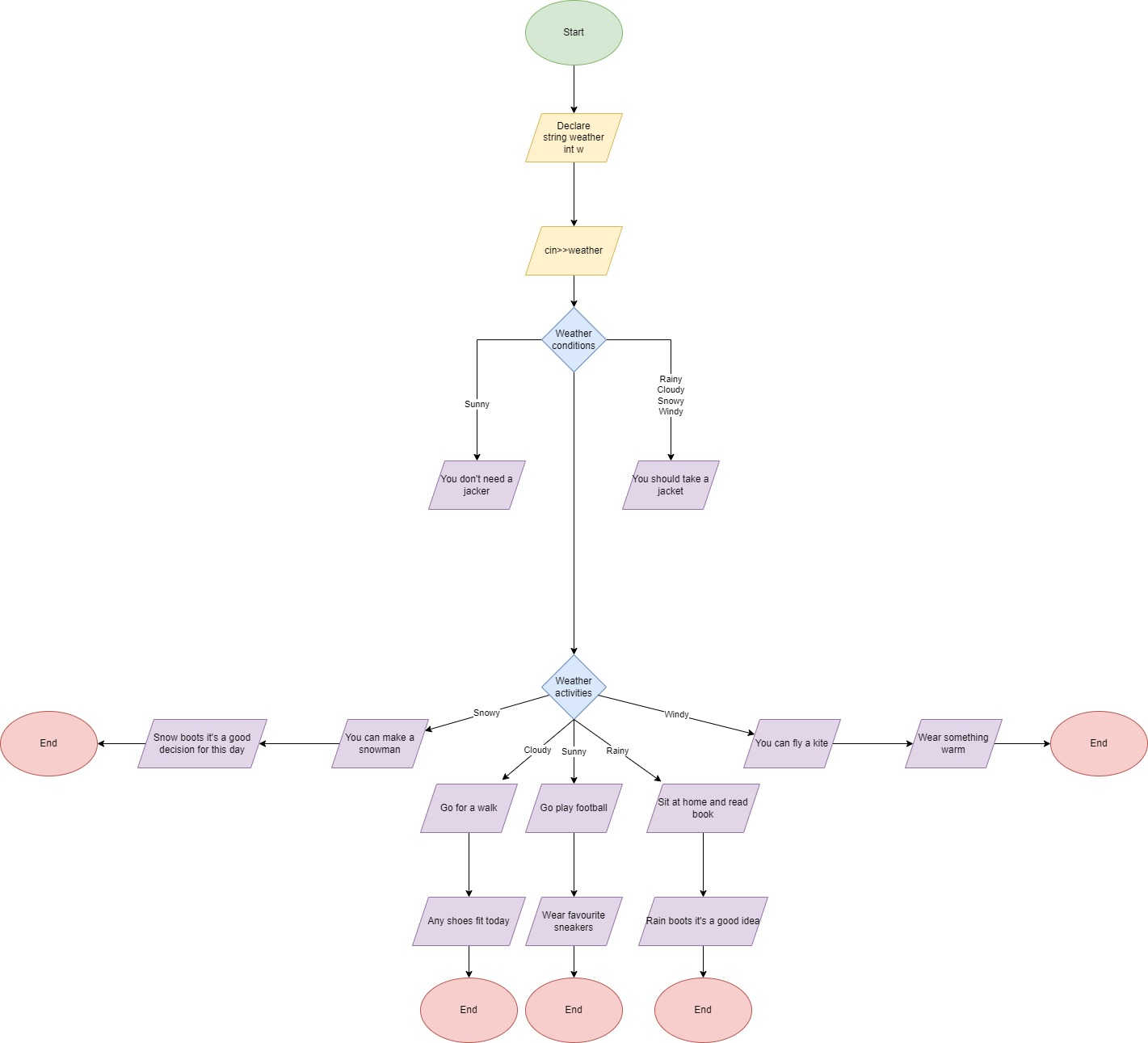


Рисунок Class Practice Task

**Програма № 4**: Self-practice task.

Блок-схема:

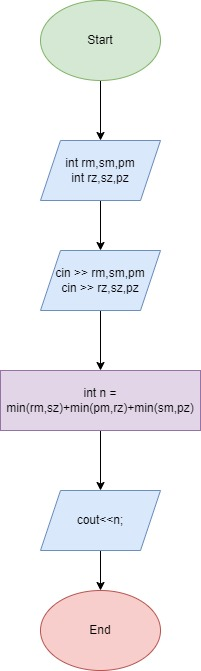


Рисунок Self-Practice Task

## **3. Код програм:**

## **Звдання №1 VNS Lab 1 Task 1.**

https://github.com/artificial-intelligence-department/ai\_programming\_playground/blob/Epic\_2\_Maksym\_Zolotarchuk/ai\_13/maksym\_zolotarchuk/Code/lab01\_task\_1.cpp

#include <iostream>

#include <cmath>

using *namespace* std;

*int* main() {

*double* a = 1000, b = 0.0001, k, p, c;

*double* k = (pow(a + b, 3) - (pow(a, 3) + 3 \* pow(a, 2) \* b));

*double* p = (3 \* a \* pow(b, 2)) + (3 \* pow(b, 3));

*double* c = k/p;

cout << "Result " << c;

return 0;

}

**https://github.com/artificial-intelligence-department/ai\_programming\_playground/blob/Epic\_2\_Maksym\_Zolotarchuk/ai\_13/maksym\_zolotarchuk/Code/lab01float.cpp**

#include <iostream>

#include <cmath>

using *namespace* std;

*int* main() {

*float* a = 1000, b = 0.0001;

*float* k = pow(a + b, 3);

*float* l = pow(a, 3) + (3 \* pow(a, 2) \* b);

*float* f = (3 \* a \* pow(b, 2));

*float* p = (3 \* pow(b, 3));

*float* c = (k - l) / (f + p);

*float* result = c;

cout << "Result " << c;

return 0;

}

**Завдання №2 Algotester Lab 1 Task 2.**

**https://github.com/artificial-intelligence-department/ai\_programming\_playground/blob/Epic\_2\_Maksym\_Zolotarchuk/ai\_13/maksym\_zolotarchuk/Code/Algotester1v2.cpp**

#include <algorithm>

#include <cmath>

using *namespace* std;

*int* main() {

*long* *long* *int* h[4];

*long* *long* *int* d[4];

*bool* turn = false;

*// Вводимо довжину ніжок*

for (*int* i = 0; i < 4; ++i) {

cin >> h[i];

}

*// Вводимо довжину відпилювання*

for (*int* i = 0; i < 4; ++i) {

    cin >> d[i];

}

*// Перевіряємо чи не відпиляємо більше довжини ніжки*

for (*int* i = 0; i < 4; ++i) {

   if( d[i] > h[i]) {

   cout << "ERROR" << endl;

   return 0;

   }

   h[i] -= d[i]; *// Відпилюємо*

*// Умова коли стіл перевернеться*

    if (\*max\_element (h, h+4) >= 2 \* (\*min\_element(h, h+4))) {

    turn = true;

    }

}

    if (!turn && \*min\_element(h, h+4) == \*max\_element(h, h+4) && \*min\_element(h, h+4) != 0) {

        cout << "YES" << endl;

    }

    else {

        cout << "NO" << endl;

    }

      return 0;

}

**Завдання № 3** **Practice# programming: Class Practice Task.**

https://github.com/artificial-intelligence-department/ai\_programming\_playground/blob/Epic\_2\_Maksym\_Zolotarchuk/ai\_13/maksym\_zolotarchuk/Code/Practice02.cpp

* #include <iostream>
* #include <string>
* using *namespace* std;
* *int* main() {
* string weather;
* *int* w;
* cout << "Enter weather:";
* cin >> weather;
* *// 1) Conditions*
* if(weather == "sunny") {
* cout << "You don't need a jacket \n";
* w = 1;
* }
* else if(weather == "rainy") {
* cout <<"You should take a jacket \n";
* w = 2;
* }
* else if(weather == "cloudy") {
* cout <<"You shouldn't take a jacket \n";
* w = 3;
* }
* else if(weather == "snowy"){
* cout <<"You should take a jacket \n";
* w = 4;
* }
* else if(weather == "windy"){
* cout <<"You should take a jacket \n";
* w = 5;
* }
* *// 2)Type of activities*
* if (weather == "sunny"){
* cout <<"You can go play football \n";
* }
* else if(weather == "rainy"){
* cout <<"Sit at home and read a book \n";
* }
* else if(weather == "cloudy"){
* cout<<"Go for a walk \n";
* }
* else if(weather == "snowy"){
* cout<<"You can make a snowman \n";
* }
* else if(weather == "windy"){
* cout <<"You can fly a kite \n";
* }
* *//3)Switch-Case*
* switch(w){
* case 1:
* cout << "Wear favourite sneakers";
* break;
* case 2:
* cout <<"Rain boots - it's a good idea!";
* break;
* case 3:
* cout<<"Any shoes fit today";
* break;
* case 4:
* cout << "Snow boots it's a good decision for this day";
* break;
* case 5:
* cout<<"Wear something warm";
* break;
* }
* return 0;
* }
* **Завдання №*4*** Self Practice Task

https://github.com/artificial-intelligence-department/ai\_programming\_playground/blob/Epic\_2\_Maksym\_Zolotarchuk/ai\_13/maksym\_zolotarchuk/Code/Self-Practice02.cpp

#include <iostream>

#include <cmath>

using *namespace* std;

*int* main() {

*int* rm,sm,pm; *// r - камінь, s - ножиці, p - папір*

cin >> rm >> sm >> pm;

*int* rz,sz,pz;

cin >> rz >> sz >> pz;

*int* n = min(rm, sz) + min(pm,rz) + min(sm,pz);

cout << n;

return 0;

}

# **Висновки:**

Під час виконання епіку 2 я ознайомився з основами синтаксису С++, змінних та констант, типами даних, видами операцій, умовними операторами, способами вводу та виводу. На практиці було реалізовано програми з використанням типів даних float та double, постфіксного та префіксного інкрементів, умовних операторів if else, if else if, switch case, cin, cout.