Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

про виконання

**Лабораторних та практичних робіт № 2**

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

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

***Виконав:***

студент групи ШІ-11

Іванов Олексій Олександрович

# **Тема роботи:**

Деталі по темі робіт

# **Мета роботи:**

Опанувати роботу з циклами та операторами галуження - основною рисою імперативних мов програмування

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Декларація змінних і констант (const/constexpr)
* Тема №2: C-масиви, C++ списки (std::vector).
* Тема №3: Умовні оператори (if, else-if, else, тернарний оператор, switch).
* Тема №4: Цикли (for, foreach, while, do while).

# **Виконання роботи:**

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

Завдання №1 VNS Lab 1 - Task 1-1

* Варіант завдання - 2
* Деталі завдання

1. Необхідно обчислити результат дробу з двома невідомими, що вводяться користувачем
2. Необхідно порівняти результати обчислень з використанням типів float і double

* Важливі деталі для врахування в імплементації програми

1. Важливо розбити приклад на вирази, і зберігати значення у змінних для уникнення помилок, зручності програміста

Завдання №2 VNS Lab 1 - Task 1-2

* Варіант завдання - 2
* Деталі завдання

1. Необхідно отримати результати дій кількох операндів на дві змінні, де може існувати двозначність
2. Необхідно дослідити дані результати і пояснити пріоритетність операндів

* Важливі деталі для врахування в імплементації програми

Існування директиви boolalpha для налаштування потоків виводу (не потрібно виводити стрічку “true” через if, можна виводити логічні змінні напряму)

Завдання № 3 Algotester Lab 1

* Варіант завдання - 2
* Деталі завдання

Дано чотири числа, які відповідають довжині ніжок столу до відпилювання. Потрібно визначити, чи можна відпиляти від кожної ніжки довжину, яка відповідає даному числу, так, щоб стіл не перевернувся і залишився паралельним підлозі.

* Важливі деталі для врахування в імплементації програми

Якщо довжина відпилювання перевищує довжину ноги, то вивести ERROR.

Завдання № 4 Class practice work

* Деталі завдання

Дано кілька видів погоди, які може ввести користувач. У залежності від вибраної погоди, необхідно дати рекомендації щодо одягу, взуття та можливих активностей для такої погоди

Завдання № 5 Self practice work

* Деталі завдання

Дано два числа, записаних у зв’язному списку по цифрах. Необхідно їх просумувати та вивести суму

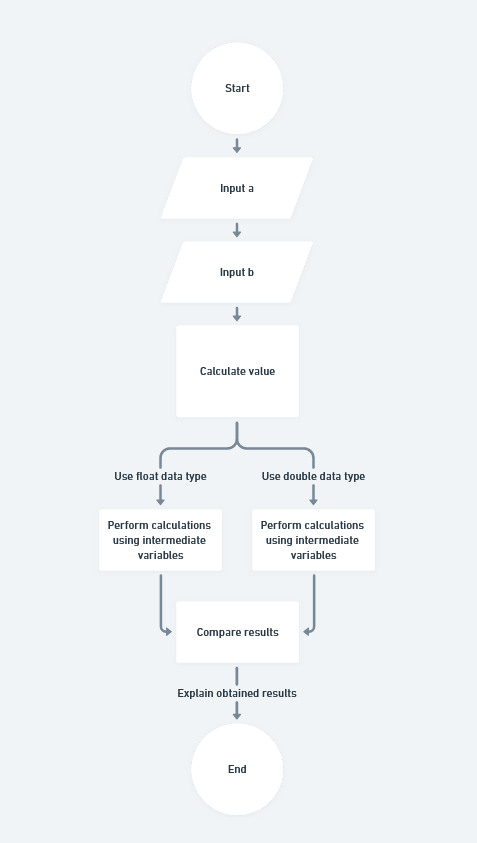
* Важливі деталі

1. Числа записані у зворотному порядку
2. Числа можуть мати різну довжину
3. Числа можуть додаватись із залишком, що переноситься наперед

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

Програма № 1 Раціональний дріб

* Блок-схема

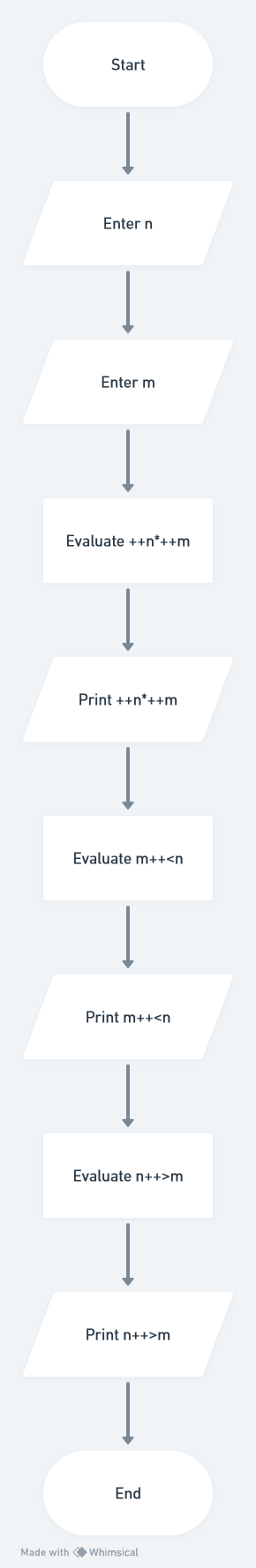


*Рисунок 3: Блок-схема до програми № 3*

* Планований час на реалізацію - 15 хвилин
* Важливі деталі для врахування в імплементації - порівняти результати обчислень при використанні float та double, не допустити перетворення знаменника в 0

Програма № 2 Оператори С++

* Блок-схема



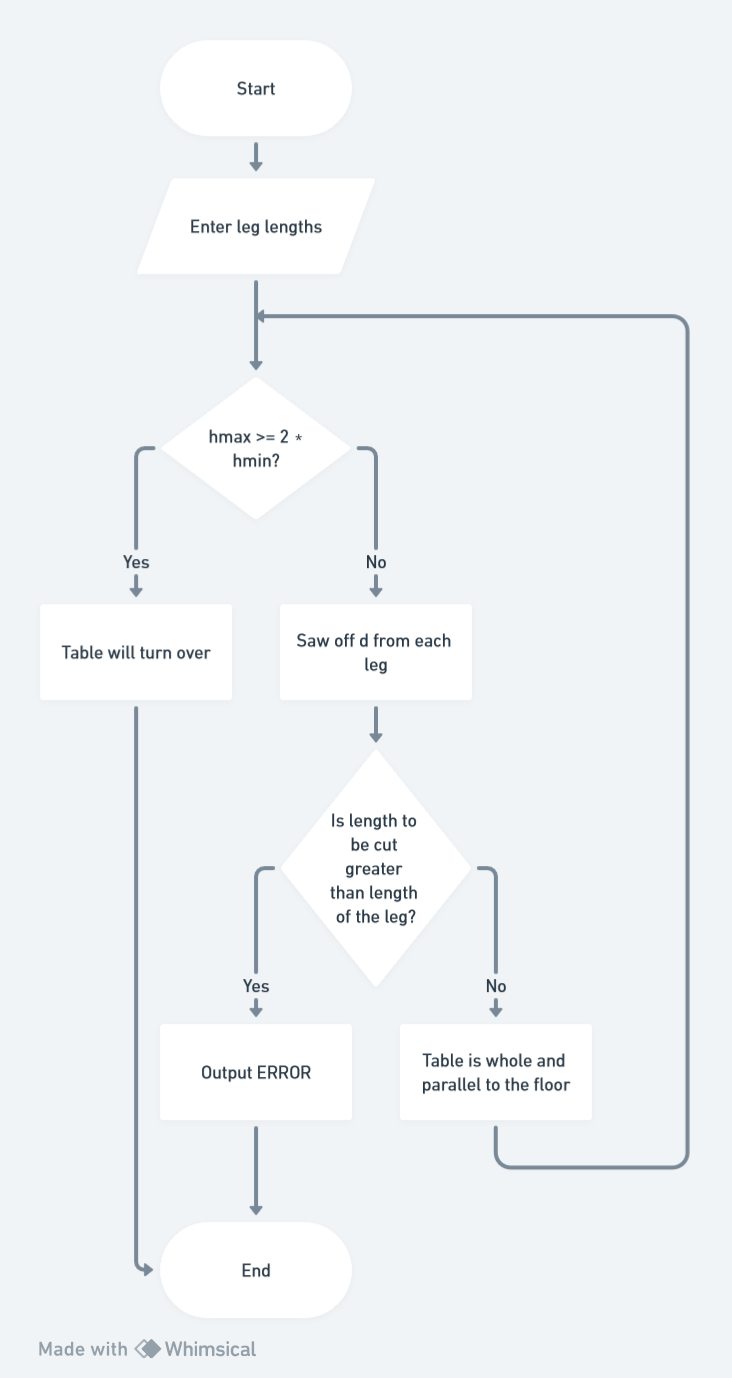
*Рисунок 2: Блок-схема до програми № 2*

* Планований час на реалізацію - 10 хвилин
* Важливі деталі для врахування в імплементації

1. Порядок виконання стандартних операторів у С++

Програма № 3 “Чи можна відпиляти чотири ніжки стола так, щоб він не перевернувся?”

* Блок-схема



*Рисунок 3: Блок-схема до програми № 3*

* Планований час на реалізацію - 20 хвилин
* Важливі деталі для врахування в імплементації (обмеження)

0 ≤ висота ніжки ≤

0 ≤ довжина спилювання ≤

Виводимо ERROR і закінчуємо програму щойно поступило хибне значення

Програма № 4 Поради по погоді

* Блок-схема



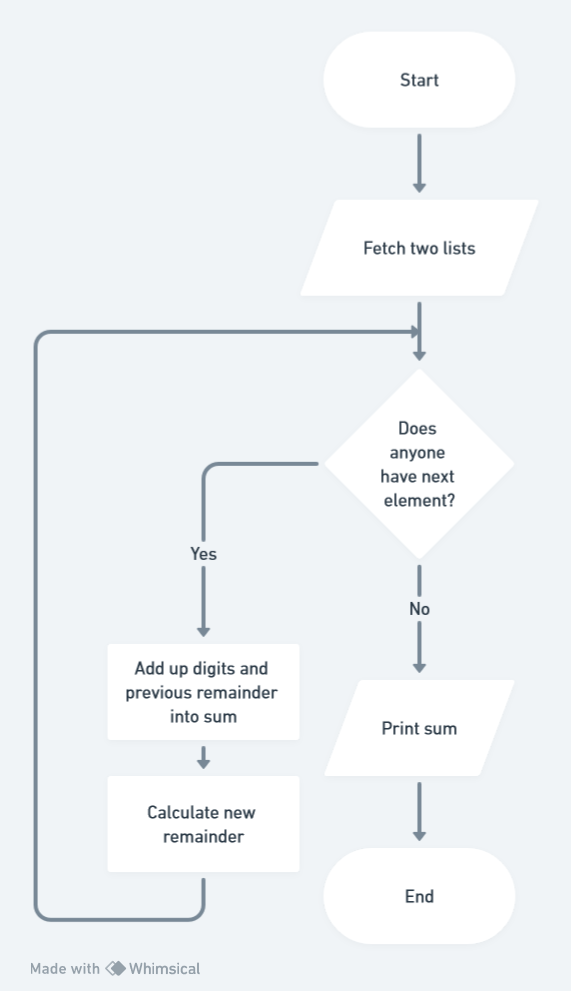
*Рисунок 4: Блок-схема до програми № 4*

* Планований час на реалізацію - 40 хвилин
* Важливі деталі для врахування в імплементації

1. Випробувати різноманітні оператори галуження (if-else, if-else if-else, switch)
2. Випробувати enum за бажанням

Програма № 5 Зворотня сума (<https://leetcode.com/problems/add-two-numbers/>)

* Блок-схема



* Планований час на реалізацію - 1 година
* Важливі деталі для врахування в імплементації

1. Перевіряти вказівники на NULL-значення

## **3. Код програм з посиланням на зовнішні ресурси:**

Завдання № 1 VNS Lab 1 - Task 1-1

#include <iostream>

#include <cmath>

using std::cout, std::cin, std::endl;

void ByDouble() {

double a = 1000, b = 0.0001;

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

denominator = pow(b, 2),

result = numerator / denominator;

cout << "The result of an expression (by doubles): " << result << "\n"; // or endl

}

void ByFloat() {

float a = 1000, b = 0.0001;

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

denominator = pow(b, 2),

result = numerator / denominator;

cout << "The result of an expression (by floats): " << result << "\n"; // or endl

}

/\*\*

\* <h4>Task 1-1</h4>

\* It resolves given mathematical expression: <p>

\* (a-b)^2 - (a^2 - 2ab) <p>

\* --------------------- <p>

\* b^2

\*/

int main() {

ByFloat();

ByDouble();

return 0;

}

[Програма № 1 (Раціональний дріб)](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/247/files#diff-7a42bdfd32ae2bc18f906557ea98c25298a80b1360a580833706bc552a00df4d)

Завдання № 2 VNS Lab 1 - Task 1-2

#include <iostream>

using std::cout, std::cin, std::endl;

*int* main() {

// Set output of booleans to its string representation

cout << std::boolalpha;

*int* n, m;

cout << "Enter n and m: ";

cin >> n >> m;

cout << "++n\*++m: " << ++n\*++m << endl;

cout << "m++<n: " << (m++<n) << endl;

cout << "n++>m: " << (n++>m) << endl;

return 0;

}

[Програма № 2 (Оператори С++)](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/247/files#diff-176ded6c5a9822e444372644c65064eae3a564cf0083576838d2a260d280bdc3)

Завдання № 3 Algotester Lab 1

#include <iostream>

#include <algorithm>

using *namespace* std;

static const *int* legsCount = 4;

*template*<*class* T>

T max(T\* *arr*, *int* *size*) {

return \*max\_element(*arr*, *arr* + *size*);

}

*template*<*class* T>

T min(T\* *arr*, *int* *size*) {

return \*min\_element(*arr*, *arr* + *size*);

}

*int* main() {

*long* *long* legsHeights[legsCount];

// reading h1, h2, h3, h4

for (*long* *long*& legsHeight : legsHeights)

cin >> legsHeight;

// reading d1, d2, d3, d4

// and sawing the leg for the di value.

*long* *long* di;

*bool* willStand = true;

for (*long* *long* & legHeight : legsHeights) {

cin >> di;

if (di > legHeight) {

cout << "ERROR";

return 0;

}

legHeight -= di;

if (max(legsHeights, legsCount)

>= 2 \* min(legsHeights, legsCount)) {

willStand = false;

}

}

cout << (willStand ? "YES" : "NO");

return 0;

}

[Програма № 3 (Відпилювання ніжок)](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/247/files#diff-165f2b7457a9383ec22930d8e93fb510f81d26ca01c79c19c2fab2b8caf0c0cf)

Завдання № 4 Поради по погоді

#include "epic2.h"

#include "helpers/string\_helper.h"

#include <iostream>

#include <vector>

#include <algorithm>

using *namespace* std;

*int* main() {

vector<string> weatherNames = { "sunny", "cloudy", "rainy", "snowy", "windy" };

string weatherStr;

cin >> weatherStr;

weatherStr = ToLowerCase(weatherStr);

*auto* it = find(weatherNames.begin(), weatherNames.end(), weatherStr);

if (it == weatherNames.end()) {

cout << "Invalid weatherStr type";

return 1; // error

}

Weather weather = (Weather)(it - weatherNames.begin());

shouldBringJacket(weather);

suggestedActivity(weather);

suggestedShoes(weather);

return 0;

}

[Програма № 4 (Поради по погоді, main.cpp)](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/247/files#diff-d7e82ff3d1fe43be81608e3c469a4764c799bfaae1c97f24c30e5ae106bfa5bf)

#ifndef epic2\_h

#define epic2\_h

*enum* Weather {

SUNNY, CLOUDY, RAINY, SNOWY, WINDY,

};

*void* shouldBringJacket(const Weather& *weather*);

*void* suggestedActivity(const Weather& *weather*);

*void* suggestedShoes(Weather *weather*);

#endif //epic2\_h

[Програма № 4 (Поради по погоді, epic2.h)](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/247/files#diff-4a05dd0fa3541eaef2b4675fb5837db993c3d65a6f25d5c7d2a399022f1cc58d)

#include <iostream>

#include "epic2.h"

using *namespace* std;

*void* shouldBringJacket(const Weather& *weather*) {

if (*weather* == Weather::SUNNY || *weather* == Weather::CLOUDY)

cout << "You can go out without a jacket";

else cout << "You should bring a jacket";

cout << endl;

}

*void* suggestedActivity(const Weather& *weather*) {

if (*weather* == Weather::SUNNY)

cout << "You may go to the park or bring some friends for picnic";

else if (*weather* == Weather::CLOUDY)

cout << "You may visit some cafes downtown or find a festival nearby";

else if (*weather* == Weather::RAINY ||

*weather* == Weather::SNOWY)

cout << "You may go to the mall, theater or a cinema, "

"or visit your friends";

else // if windy

cout << "You may bring up a kite to play with it outdoors";

cout << endl;

}

*void* suggestedShoes(Weather *weather*) {

switch (*weather*) {

case Weather::SUNNY:

cout << "If you go out, put on sneakers";

break;

case Weather::CLOUDY:

cout << "If you go out, put on any shoes you like";

break;

case Weather::RAINY:

cout << "If you go out, put on water-repellent shoes";

break;

case Weather::SNOWY:

cout << "If you go out, put on boots";

break;

case Weather::WINDY:

cout << "If you go out, put on leather shoes";

break;

default:

break;

}

cout << endl;

}

[Програма № 4 (Поради по погоді, epic2.cpp)](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/247/files#diff-71f790d6d49e123e6d1d2018f6728d5e78158040e53b2269816c6d31f34102c5)

Завдання № 5 Зворотня сума

/\*\*

\* Definition for singly-linked list.

\* struct ListNode {

\* int val;

\* ListNode \*next;

\* ListNode() : val(0), next(nullptr) {}

\* ListNode(int x) : val(x), next(nullptr) {}

\* ListNode(int x, ListNode \*next) : val(x), next(next) {}

\* };

\*/

class Solution {

public:

ListNode\* addTwoNumbers(ListNode\* l1, ListNode\* l2) {

// holding linked list pointer and current node of it

ListNode\* new\_list = new ListNode, \*new\_node = new\_list;

// holding remainder of addition

int remainder = 0;

// writing current node if any of lists have any members

while (l1 != nullptr || l2 != nullptr) {

// summing everything up

int sum = remainder;

if (l1 != nullptr) sum += l1->val;

if (l2 != nullptr) sum += l2->val;

// writing last digit as value

new\_node->val = sum % 10;

// evaluating remainder

remainder = sum / 10;

// if there is next value in any of lists,

if (

(l1 != nullptr && l1->next != nullptr) ||

(l2 != nullptr && l2->next != nullptr)

) {

// next node is needed

new\_node->next = new ListNode;

}

// or if there is no elements, but remainder isn't equal to 0

else if (remainder != 0) {

new\_node->next = new ListNode(remainder);

}

// moving iterating pointers to next positions

if (l1 != nullptr) l1 = l1->next;

if (l2 != nullptr) l2 = l2->next;

new\_node = new\_node->next;

}

// returning resulting list

return new\_list;

}

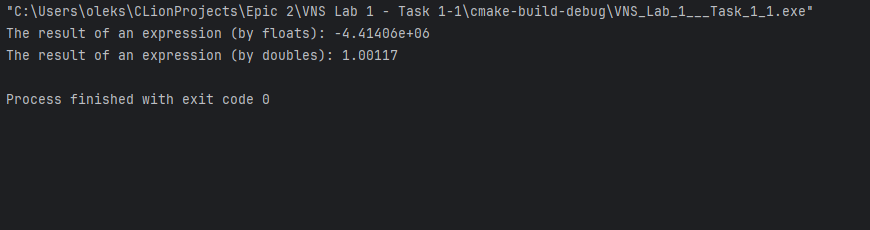
};

[Програма № 5 (Зворотна сума)](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/247/files#diff-312ba525c643d73b417c391aa452d20d33acf0b203787ca0a91d93e0ba7e932f)

[**Посилання на pull request**](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/247)

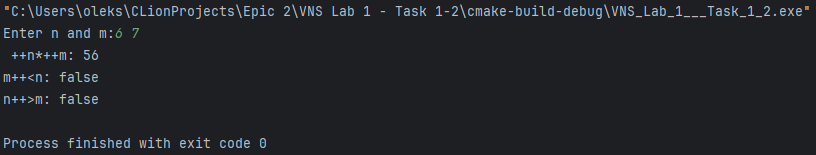
## **4. Результати виконання завдань, тестування та фактично витрачений час:**

Завдання № 1



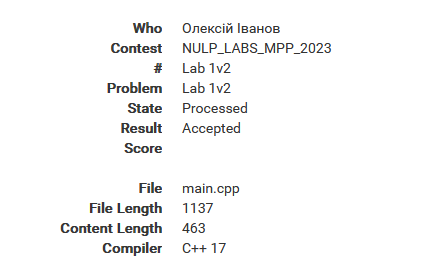
Час, витрачений на виконання завдання - 15 хвилин

Завдання № 2



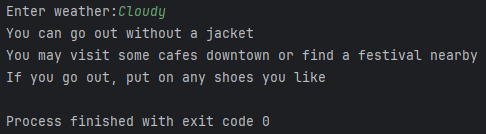
Час, витрачений на виконання завдання - 10 хвилин

Завдання № 3



Час, витрачений на виконання завдання - 20 хвилин

Завдання № 4



Час, витрачений на виконання завдання - 50 хвилин

Завдання № 5



Час, витрачений на виконання завдання - 1 година

# **Висновки:**

В даному циклі лабораторних робіт я опанував роботу з циклами for, while та do-while; операторами галуження - if, else if, else, switch, goto; навчився застосовувати їх для вирішення різноманітних задач.