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 алгоритми. Розгалужені алгоритми. Умовні оператори. Змінні. Константи. Операції.»

***Виконав:***

студент групи ШІ-11

Ковалець Владислав Миколайович

# **Тема роботи:**

" Лінійні алгоритми. Розгалужені алгоритми. Умовні оператори. Змінні. Константи. Операції"

# **Мета роботи:**

Ознайомитися з лінійними і розгалуженими алгоритмами та їх застосуванням. Розглянути умовні оператори та способи задання різних типів змінних.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Лінійні та розгалужені алгоритми.
* Тема №2: Умовні оператори.
* Тема №3: Константи та змінні.

1. Індивідуальний план опрацювання теорії:

* Тема №1: Лінійні та розгалужені алгоритми.

Джерела Інформації:

* + - Відео:

https://www.youtube.com/watch?v=EPfSVJkjx\_0

* + Що опрацьовано:
    - Основні методи побудови алгоритмів
  + Статус: Ознайомлений
  + Початок опрацювання теми: 10.10.2023
  + Звершення опрацювання теми: 11.10.2023
* Тема №2: Умовні оператори.
  + Джерела Інформації:
    - Стаття.

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

* + Що опрацьовано:
    - Розглянуто оператори умовного розгалуження if/else та те, як їх можна використовувати.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 10.10.2023
  + Звершення опрацювання теми: 14.10.2023
* Тема №3: Константи та змінні.
  + Джерела Інформації:
    - Стаття.

http://cpp.dp.ua/konstanty/

* + Що опрацьовано:
    - Поняття змінних та констант та їх ведення
  + Статус: Ознайомлений
  + Початок опрацювання теми: 10.10.2023
  + Звершення опрацювання теми: 10.10.2023

# **Виконання роботи:**

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

Завдання № VNS lab1

* 24 варіант

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

2. Обчислити значення виразів. Пояснити отримані результати.

Важливі деталі для врахування в імплементації програми:

-використати float й double і пояснити значення результатів.

- пояснити отримані результати.

Завдання № algotester lab 1

* 3 варіант
* Персонажу по одному дають сторони 5 кубiв a1..5, з яких вiн будує пiрамiду.

Коли вiн отримує куб з ребром ai - вiн його ставить на iснуючий, перший ставить на пiдлогу (вона безмежна). Якщо в якийсь момент об’єм куба у руцi (який будуть ставити) буде бiльший нiж у куба на вершинi пiрамiди - персонаж програє i гра закiнчується. Розмiр усiх наступних кубiв пiсля програшу не враховується. Тобто якщо ai−1 < ai - це програш. Ваше завдання - сказати як закiнчиться гра.

* **Важливі деталі для врахування в імплементації програми:**
* Вхiднi данi

5 цiлих чисел a1..a5 - сторони кубiв

* Вихiднi данi

Iснуючi варiанти:

LOSS - якщо персонаж не зможе поставити куб.

WIN - якщо персонаж зможе поставити усi куби

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

* Обмеження

−10^12 ≤ ai ≤ 10^12

Завдання № практична №2

* Загальний варіант

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.

* **Важливі деталі для врахування в імплементації програми:**

Використовувати дані цикли.

Завдання № Self Practice Work

* Варіант –
* Зібралися Зеник і Марічка разом з пластунами в похід. Похід — серйозна справа. Потрібно запастись продуктами харчування та розподілити їх споживання по днях так, щоб всім вистачило. Цього разу Зеник слідкує за тим, щоб печива вистачило аж до останнього дня походу. Зеник чітко знає, скільки пачок печива повинно залишитись кожного дня, і щовечора перераховує їх. Якщо Зеник побачить, що залишилось менше пачок, ніж повинно залишитись за його розрахунками, він неодмінно знайде того, хто з’їв забагато печива, і покарає його.

Марічка дуже любить печиво. Сьогодні, коли всі пластуни покинуть свої намети і підуть купатися в річку, Марічка планує непомітно з’їсти трохи печива. Звісно, Марічка не хоче бути покараною і дуже боїться, щоб Зеник не помітив пропажу.

Марічка підгледіла, скільки пачок печива є в рюкзаку Зеника. Також вона знає, скільки штук в кожній пачці. Марічці не терпиться дізнатися, скільки ж печива вона зможе з’їсти так, щоб Зеник не помітив. Зеник помітить пропажу печива з деякої пачки тоді і тільки тоді, коли Марічка повністю спустошить її.

* **Важливі деталі для врахування в імплементації програми:**

Вхідні дані

У першому рядку задано одне натуральне число n — кількість пачок печива.

У другому рядку задано n натуральних чисел a(i) — кількість штук печива в пачці.

Вихідні дані

У єдиному рядку виведіть одне ціле число — максимальну кількість штук печива, яку зможе з’їсти Марічка так, щоб Зеник не помітив цього.

Обмеження

20% тестів:

1≤n≤1000,0≤a(i)≤10^4

60% тестів:

1≤n≤105,0≤a(i)≤10^4

20% тестів:

1≤n≤105,0≤a(i)≤10^9

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

Програма № VNS lab1

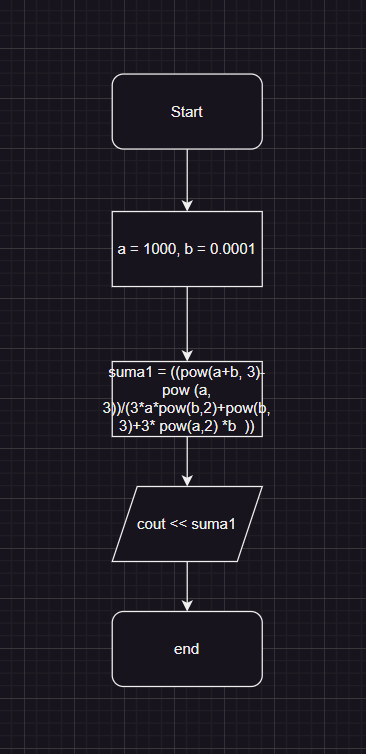
* Блок-схема:
* 

Рисунок Блок-схема до VNS lab1 блок №1, 2

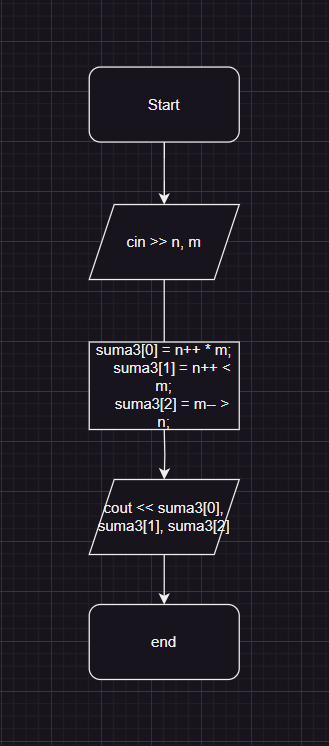
* 

Рисунок Блок схема до VNS lab1 блок №3

* Планований час на реалізацію

10 хвилин

Програма № algotester lab1

* Блок-схема

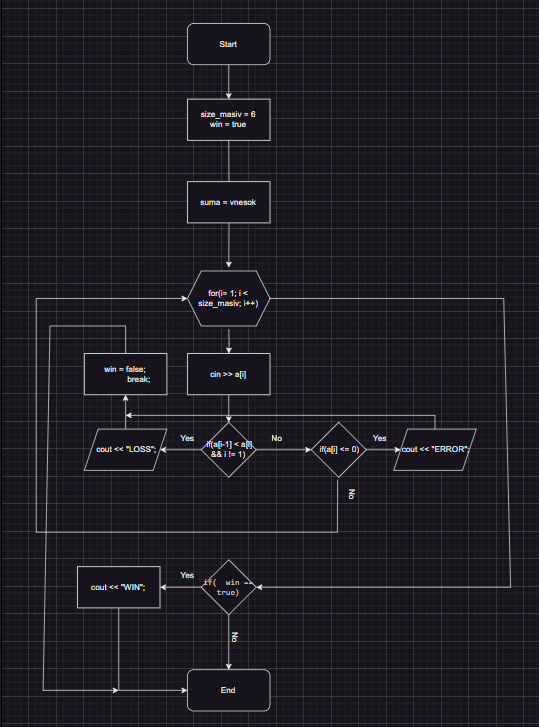


Рисунок Блок-схема до algotester lab1

* Планований час на реалізацію

1 годину

* Важливі деталі для врахування в імплементації

Програма № практика №2

* Блок-схема

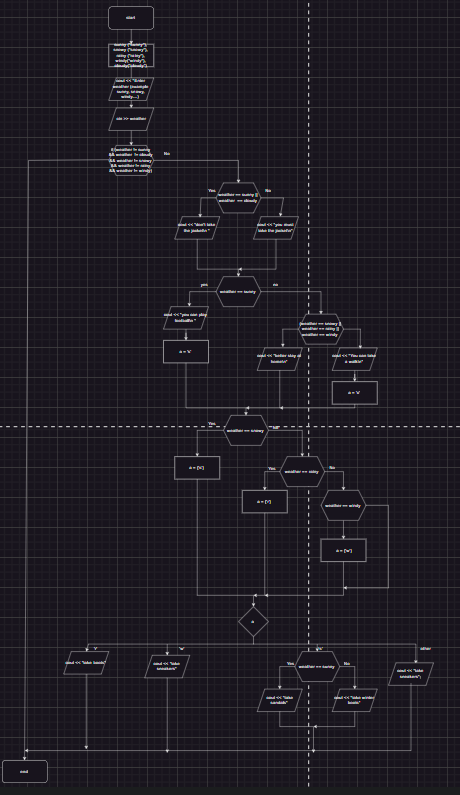


Рисунок Блок-схема до практичної роботи

* Планований час на реалізацію

1 година

Завдання № Self Practice Work

* Блок схема

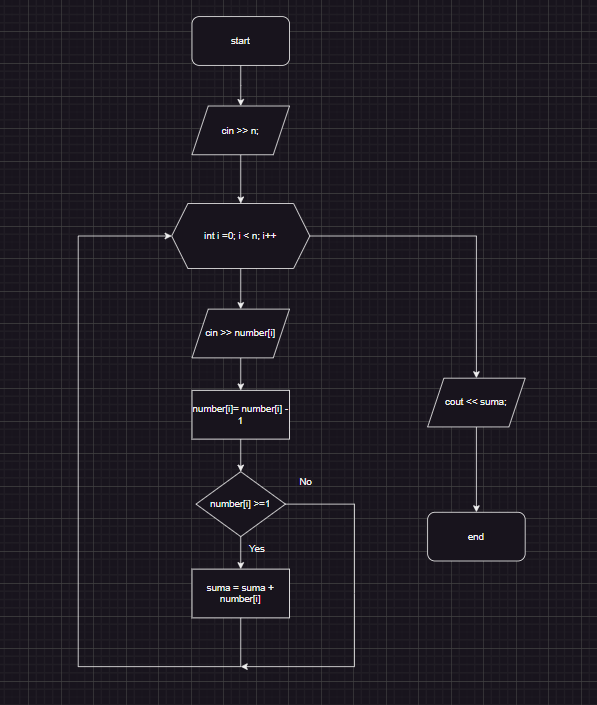


Рисунок Блок схема до завдання self practice

## **3. Конфігурація середовища до виконання завдань:**

Завдання № - Деталі по конфігурації середовища + скріншоти з підписами до скріншотів.

**Операційна система**: Windows

**Інсталяція інтерпретаторів або компіляторів: С++**

**Інсталяція залежностей: -**

**Інтегроване середовище розробки (IDE)**: Visual Studio Code

**Зовнішні ресурси**: -

**Завантажені бібліотеки або плагіни**:



Рисунок Завантажені бібліотеки або плагіни

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

**Завдання № VNS1**

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

2. Обчислити значення виразів. Пояснити отримані результати

#include <iostream>

#include <math.h>

int main()

{

    double a = 1000, b = 0.0001;

    double suma1;

    suma1 = ((pow(a+b, 3)- pow (a, 3))/(3\*a\*pow(b,2)+pow(b, 3)+3\* pow(a,2) \*b  ));

    std::cout << "Pruklad = " << suma1 << "\n";

    return 0;

}

#include <iostream>

#include <math.h>

int main()

{

    float a = 1000, b = 0.0001;

    float suma1;

    suma1 = ((pow(a+b, 3)- pow (a, 3))/(3\*a\*pow(b,2)+pow(b, 3)+3\* pow(a,2) \*b  ));

    std::cout << "Pruklad = " << suma1 << "\n";

    return 0;

}

#include <iostream>

int main()

{

    int suma3[3];

    int  n, m;

    std::cin >> n;

    std::cin >> m;

    suma3[0] = n++ \* m;

    suma3[1] = n++ < m;

    suma3[2] = m-- > n;

    std::cout << suma3[0]<< std::endl << suma3[1] << std::endl << suma3[2];

    return 0;

}

**Завдання № algotester lab1**

1. Розв’язати задачу

#include <iostream>

using namespace std;

int main()

{

    int size\_masiv = 6;

    long long a[size\_masiv];

    bool win = 1;

    for (int i = 1; i < size\_masiv; i++)

    {

        cout << "Enter size side";

        cin >> a[i];

        if(a[i-1] < a[i] && i != 1)

        {

            cout << "LOSS";

            win = false;

            break;

        }

        if(a[i] <= 0)

        {

            cout << "ERROR";

            win = false;

            break;

        }

    }

    if(  win == true)

    cout << "WIN";

    return 0;

}

**Завдання № практика 2**

#include <iostream>

#include <string>

using namespace std;

int main ()

{

    string weather;

    string sunny ("sunny"), snowy ("snowy"), rainy ("rainy"), windy("windy"), cloudy("cloudy");

    cout << "Enter weather (example sunny, snowy, windy....) ";

    cin >> weather;

    char a;

    if (weather != sunny && weather  != cloudy && weather != snowy && weather != rainy && weather != windy)

    {

        cout << "weather is undefiend";

    }

    else

    {

        if (weather == sunny || weather  == cloudy)

        {

            cout << "don't take the jacket\n ";

        }

            else

            {

                cout << "you must take the jacket\n";

            }

        if (weather == sunny)

        {

            cout << "you can play football\n ";

            a = 's';

        }

            else if (weather == snowy || weather == rainy || weather == windy )

            {

                cout << "better stay at home\n";

            }

            else

            {

                cout << "You can take a walk\n";

                a = 'c';

            }

        if (weather == snowy)

            a = {'s'};

        else if (weather == rainy)

            a = {'r'};

        else if (weather == windy)

            a = {'w'};

        switch (a)

        {

            case 'r':

            cout << "take boots";

            break;

            case 'w':

            cout << "take sneakers";

            break;

            case 's':

            if (weather == sunny)

            cout << "take sandals";

            else

            cout << "take winter boots";

            break;

            default:

            cout << "take shoes";

            break;

        }

    }

    return 0;

}

**Завдання № Self Practice Work**

1. Розв’язати задачу

#include <iostream>

using namespace std;

int main()

{

    unsigned long n, suma = 0;

    unsigned long number[100000];

    cin >> n;

    for(int i=0; i< n; i++)

    {

        cin >> number[i];

        number[i] = number[i] - 1;

        if (number[i] >= 1)

        {

            suma = suma + number[i];

        }

    }

    std::cout << suma;

}

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

Завдання № vns lab1

1.При заданні змінних float значення суми рівне = 1.2207 (блок 1), а при змінних double сума = 1 (блок 2)

2. (Блок 3) При заданні n і m довільними числами( наприклад 8 і 8 ), перший приклад буде = 64, тому що n збільшується на одиницю після множення, тому що пріоритет n++ менший за інші. В другому прикладі значення буде false, бо буде 9 < 8, бо n перед тим збільшиться на 1 і після обрахунку збільшиться на 1. Третій приклад буде false, тому що буде 8 > 10, а m зменшиться вже після обрахунку.

Час затрачений на виконання завдання

20 хвилин

Завдання № algotester lab1

Програма призначена для визначення результату гри, де гравець будує піраміду з п'яти кубів. У разі правильної побудови програма виводить "WIN." У разі порушення правил гри, наприклад, коли розмірність куба менше попереднього або введено невірні дані, програма виводить "LOSS" або "ERROR".



Рисунок Підтвердження перевірки коду на algotester

Час затрачений на виконання завдання

20 хвилин

Завдання № практична 2

Програма програма призначена для визначення, що одягнути. Вона спрацьовує наступним чином: Користувач вводить погоду, наприклад, "sunny," "snowy," "windy," і т.д. Програма перевіряє введену погоду і робить висновок щодо того, який одяг підходить до цієї погоди. Вона використовує рядок для введення погоди, і ви вже маєте підготовлені рядки, що відповідають різним погодним умовам. Використовується if-else конструкції для визначення типу одягу на основі введеної погоди. Крім того, використовується switch для вибору взуття (ботінки, кросівки тощо) в залежності від погоди та типу взуття.

Час затрачений на виконання завдання

20 хвилин

Завдання № Self Practice Work

У ході виконання програми, яка моделює ситуацію з подорожжю Зеника, Марічки та пластунів, було отримано результати, які дозволяють визначити максимальну кількість штук печива, яку Марічка може з'їсти, щоб Зеник не помітив пропажу. Результати виконання програми свідчать про те, що програма обчислює все правильно.



Рисунок Підтвердження перевірки коду на algotester

Час затрачений на виконання завдання

20 хвилин

# **Висновки:**

Було розглянуто різні типи операторів, такі як умовні оператори (if-else, switch-case) та додатково було розглянуто циклічні оператори (for), на практиці навчилися їх використовувати для розв'язання завдань умова, яких складніша за прості лінійні задачі. Протягом практичних завдань нам вдалося розробляти програми, які взаємодіють з користувачем, вирішувати складні завдання та приймати рішення на основі різноманітних умов. Такий підхід дозволяє ефективно використовувати програмування для вирішення реальних завдань та автоматизації процесів.