Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

про виконання

**Лабораторних та практичних робіт**

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

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

***Виконав:***

студент групи ШІ-12

Ширяєв Іван Павлович

# **Тема роботи:**

Ознайомлення з базовими поняттями про змінні, константи та умовні оператори. Алгоритми. Виконання різноманітних завдань по створенню лінійних та розгалужених алгоритмів на мові C/C++ .

# **Мета роботи:**

Навчитись створювати алгоритми різних видів. Навчитись використовувати умовні оператори для написання розгалужених алгоритмів. Практика в написанні програм різної структури на C/C++.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Знайомство з С. Виконання програми простої структури.
* Тема №2: Використання основних операторів мови С

1. Індивідуальний план опрацювання теорії:

* Тема №1: Знайомство з С. Виконання програми простої структури.
  + Джерела Інформації
    - <https://www.prepbytes.com/blog/c-programming/structure-of-c-program-with-example/>
    - <https://www.geeksforgeeks.org/structure-of-c-program/>
    - <https://www.youtube.com/watch?v=U3aXWizDbQ4>
  + Що опрацьовано:
    - Опрацьовано структуру програми в C/C++
    - Константи, змінні, базові операції
  + Статус: Ознайомлений
  + Початок опрацювання теми: 20.10.2023
  + Звершення опрацювання теми: 21.10.2023
  + Тема №2: Використання основних операторів мови С
  + Джерела Інформації:
    - <https://www.geeksforgeeks.org/operators-in-c/>
    - <https://www.w3schools.com/cpp/cpp_for_loop.asp>
    - <https://en.cppreference.com/w/cpp/language/statements>
  + Що опрацьовано:
    - Опрацьовано складені оператори, оператори вибору, оператори циклу і оператори переходу (for,while, if\if else, switch case, break, continue)
  + Статус: Ознайомлений
  + Початок опрацювання теми: 20.10.2023
  + Звершення опрацювання теми: 22.10.2023

# **Виконання роботи:**

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

Завдання VNS lab 1 tasks 1, 2

* Варіант завдання 25
* Деталі завдання

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

(2) Потрібно обчислити значення виразів. Пояснити отримані результати.

* Важливі деталі для врахування в імплементації програми

(1) При обчисленні використовувати проміжні змінні

Завдання VNS lab 2 task 1

* Варіант завдання 1
* Деталі завдання

Знайти суму цілих додатніх чисел, кратних 3 і менших 200.

* Важливі деталі для врахування в імплементації програми

Використовуємо int для знаходження кількості чисел кратних 3 з 200 бо він відкидає дробову частину.

Завдання Algotester Lab 1

* Варіант завдання 3
* Деталі завдання

Потрібно вивести результат відповідно вхідним даним, тобто якщо введений користувачем ряд зменшувальний або складається з однакових чисел - “перемога”, інакше - “поразка”, якшо користувач вводить недопустимі значення - “помилка”

* Важливі деталі для врахування в імплементації програми

Використано bool та оператор &&, використовуємо long long для елементів масиву бо обмеження −10^12 ≤ ai ≤ 10^12

Завдання Algotester Lab 2

* Варіант завдання 1
* Деталі завдання

У вас є дорога, яка виглядає як N чисел.

Пiсля того як ви по нiй пройдете - вашу втому можна визначити як рiзницю максимального

та мiнiмального елементу.

Ви хочете мiнiмiзувати втому, але все що ви можете зробити - викинути одне число з дороги,

тобто забрати його з масиву.

В результатi цiєї дiї, яку мiнiмальну втому ви можете отримати в кiнцi дороги?

* Важливі деталі для врахування в імплементації програми

Використано bubble sort та функцію min, для даних використовуємо int бо великі значення не потрібні

Завдання Algotester self practice

* Варіант завдання відсутній
* Деталі завдання

Вам задано послідовність із n цілих чисел ai. Ваша задача — знайти довжину найбільшої зростаючої підпослідовності заданої послідовності.

* Важливі деталі для врахування в імплементації програми

Використано функцію max, важливо щоб ряд в який ми заносимо довжину послідовностей складався з одиниць.

Practice work team task

* Варіант завдання відсутній
* Деталі завдання

Задано декілька типів погоди, потрібно видати рекомендації щодо активностей на основі введеної погоди.

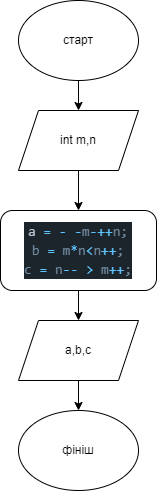
* Важливі деталі для врахування в імплементації програми

Потрибно додати бібліотеку <string >

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

Програма VNS lab 1 tasks 1, 2

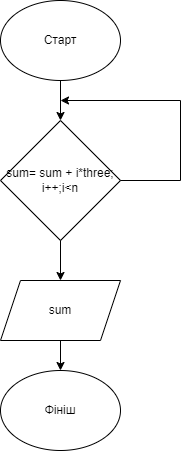
* Блок-схема



* Планований час на реалізацію: 20 хвилин на кожне

Програма № VNS lab 2 task 1

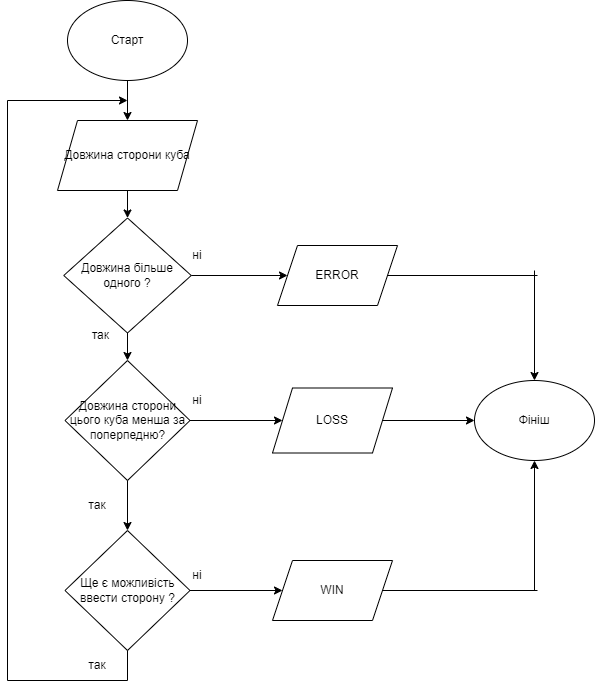
* Блок-схема



* Планований час на реалізацію 30 хвилин

Програма № Algotester Lab 1

* Блок-схема

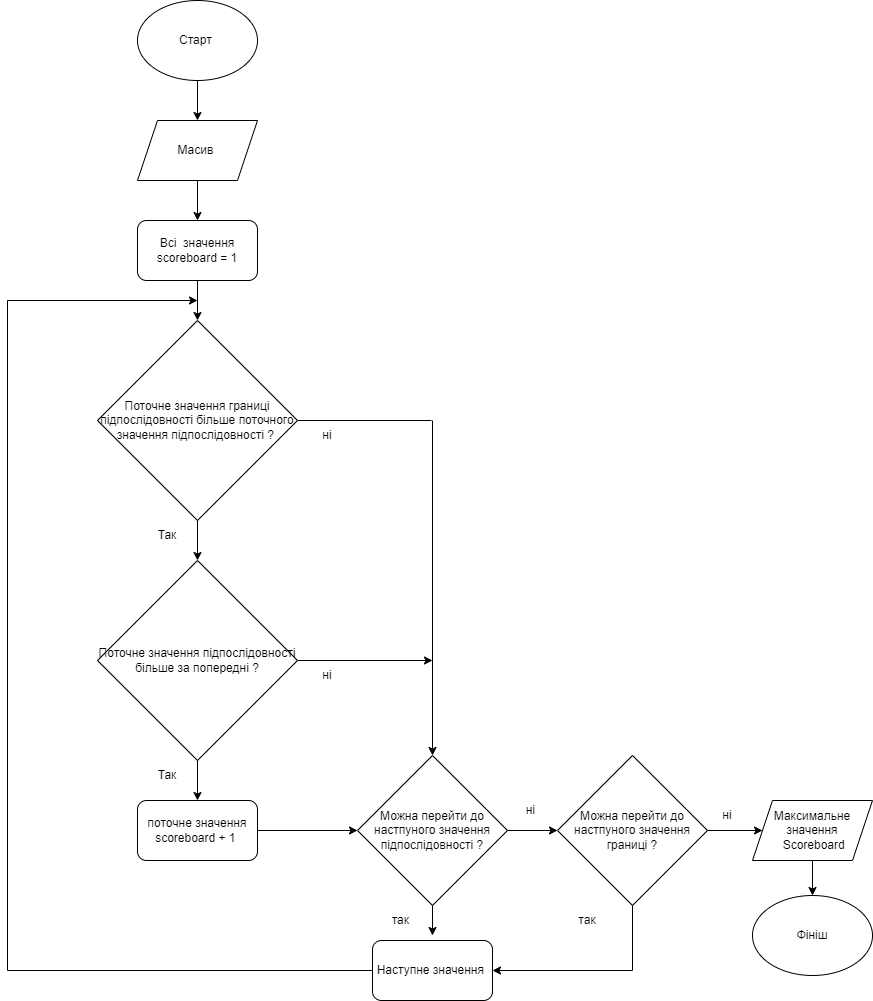


* Планований час на реалізацію 1 день
* Програма № Algotester Lab 2



- Планований час на реалізацію 1 день

* Програма № Algotester self practice



- Планований час на реалізацію 1 день

* Practice work team task



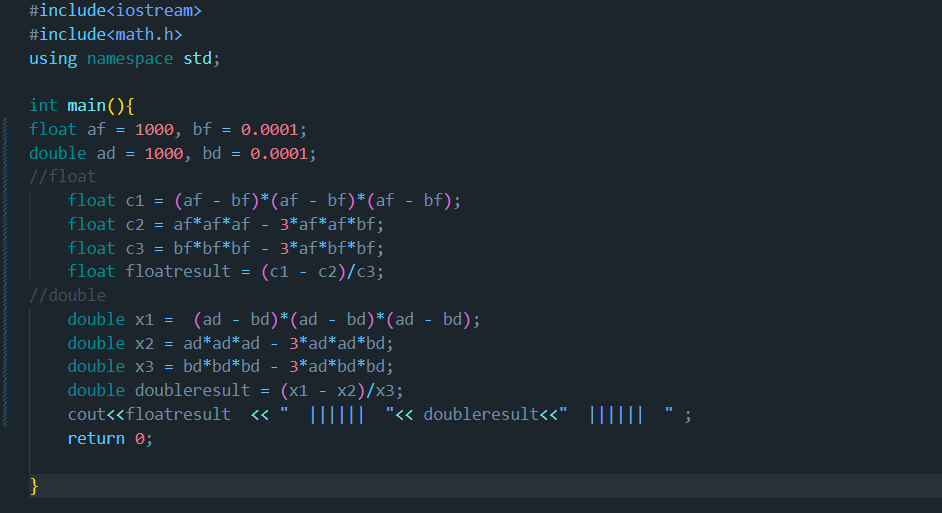
- Планований час на реалізацію 2 години

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

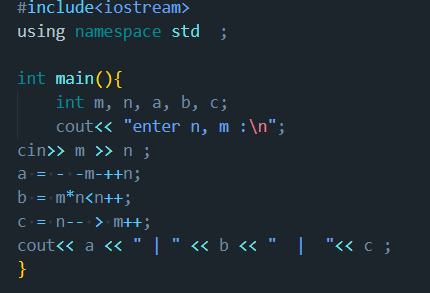
Додаткова конфігурація не потрібна.

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

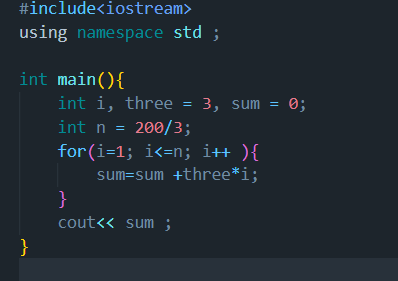
Завдання VNS lab 1 tasks 1

 Код VNS lab 1 task 1

Завдання VNS lab 1 tasks 2

Код VNS lab 1 task 2

Завдання VNS lab 2 task 1

 Код VNS lab 2 task 1

Знаходимо кількість чисел кратних 3 з 200, це 200/3, через те що використано тип даних int дробова частина відкидається і отримуємо ціле число, потім за допомогою циклу for знаходимо суму всіх цих чисел.

Завдання Algotester Lab 1

Algotester lab 1

Створюємо цикл в середині якого ми вводимо значення масив у і одразу ж виконуємо всі необхідні перевірки, якщо введене значення менше 1 то виводимо “ERROR”, потім якщо введене число більше попереднього виводимо “LOSS”(тільки для чисел з індексом більше 0), якщо всі числа пройшли перевірку то виводимо “WIN”

Завдання Algotester lab 2

 Algotester lab 2

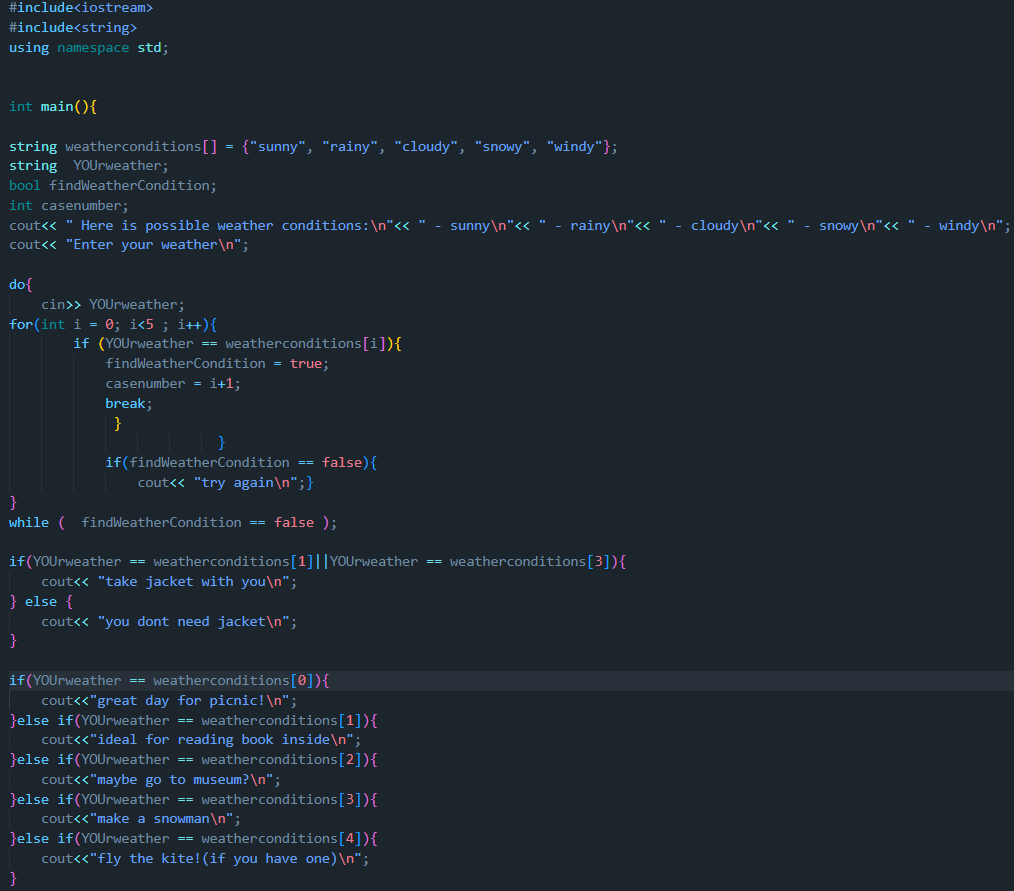
В завданні не зазначено чи обов’язково потрібно вводити значення масиву у збільшувальному порядку, тому я використав bubble sort для того щоб відсортувати значення та працювати з ними більш просто. Нам потрібно “викинути” один елемент з масиву для того щоб знайти мінімальну втому, “викидувати” будемо тільки перший або останній елемент масиву бо тільки в цьому є сенс. Тобто розглядаємо два випадки: різниця першого і передостаннього елементу масиву і навпаки - другого і останнього, обчислюємо їх за допомогою функції min знаходимо менше з цих двох значень, виводимо його. Також для окремого випадку коли довжина дороги буде 1 виводимо 0.

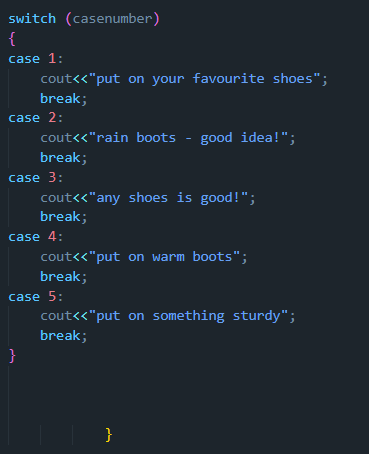
Завдання Algotester self practice

Algotester self practice

Спочатку створюємо цикл який перетворить всі елементи масиву на одиниці, це потрібно через те що мінімальною підпослідовністю є одне число, тобто мінімальна довжина 1. Після цього створюємо цикл який розглядає “підпослідовності” масиву перший член яких є array[j] а останній array[i], ми послідовно порівнюємо останній член з кожним з членів перед ним, якщо array[i] > array[j] заносимо довжину підпослідовності в відповідну “комірку” масиву scoreboard, для цього використовуємо функцію max яка обирає більше з двох значень, порівнюємо саме scoreboard[i] та scorebord[j] + 1 через те що ми не знаємо чи більше поточне значення array[j] за попереднє, тому якщо це правда то ми мали занести його в scoreboard до цього моменту тому scoreboard[j] + 1 буде більше за scoreboard[i] і це значення ми занесемо до нової комірки, якщо це інакше то ми пропускаємо його. Після цього ми шукаємо найбільше значення в scoreboard і виводимо його.

Завдання Practice work team task

 Practice work team task

Фігура 1: Practice work team task

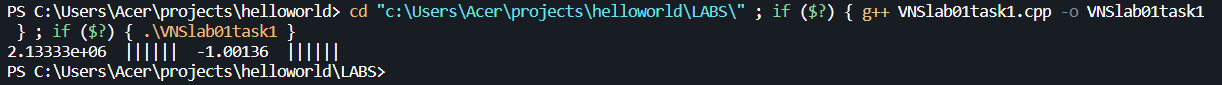
Для того щоб з’ясувати чи є введена погода в списку створюємо while цикл що послідовно порівнює введений string зі всіма іншими(під час цього назначаємо значення що буде відповідати відповідному switch case), якщо це правда - булєве значення true, інакше - false, просимо ввести користувача правильну погоду знову.Використовуємо 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.

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

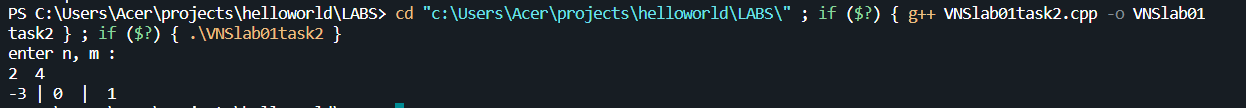
Завдання VNS lab 1 tasks 1 Деталі по виконанню і тестуванню програми

 VNS lab 1 task 1 results

Програмам повертає різні значення через те що ми вказали різні типи даних .

Час затрачений на виконання завдання 20 хвилин

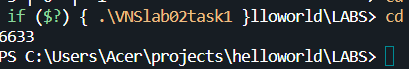
Завдання VNS lab 1 tasks 2

 VNS lab 1 task 2 results

Вводимо значення перше - віднімає від n m+1, друге - порівнняння m\*n і n+1(повертає 1 або 0), третє - так само тільки порівнює n-1 і m+1.

Час затрачений на виконання завдання 20 хвилин

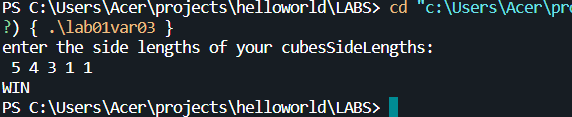
Завдання VNS lab 2 task 1

VNS lab 2 task 1 results

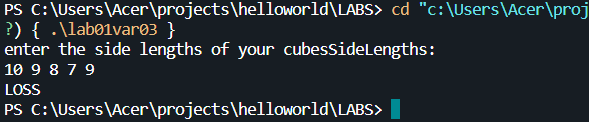
Повертає суму всіх чисел кратних 3 менших 200.

Час затрачений на виконання завдання 1 година

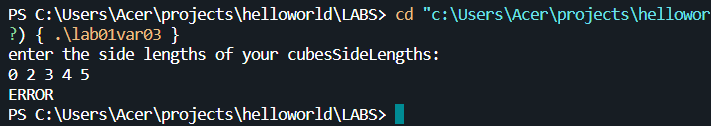
Завдання Algotester Lab 1

 ALgotester lab 1 results

Ситуація “Перемоги”

 Algotester lab 1 results

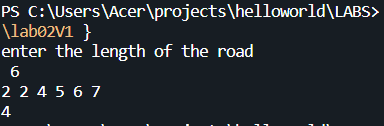
Ситуація “Поразки”

 ALgotester lab 1 results

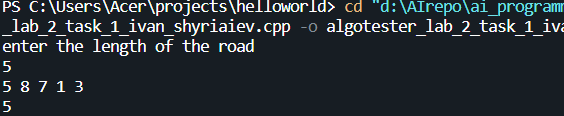
Ситуація “Помилки”

Час затрачений на виконання завдання 1 день

Завдання Algotester lab 2

Algotester lab 2 results

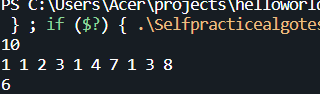
Перший рядок - довжина масиву, другий - його елементи, третє “мінімальна втома”

 Algotester lab 2 results

Якщо масив не відсортований

Час затрачений на виконання завдання 1 день

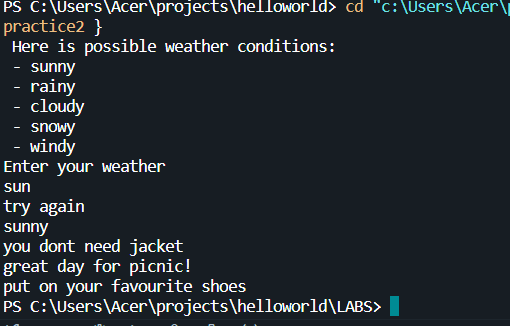
Завдання Algotester self practice

 Algotester self practice results

Тут найбільша зростаюча послідовність це [1,2,3,4,7,8]

Час затрачений на виконання завдання 5 днів

Завдання Practice work team task

 Practice work team task results

Час затрачений на виконання завдання 2 години

# **Висновки:**

Я ознайомився з базовими поняттями про змінні, оператори та алгоритми, вивчив базову структуру програми в C\C++. Більш детально розглянув тему операторів умови та розгалуження, в завданях потрібно було використати for, if(if else), do while, switch case, також я використовував такі логічні оператори як || та && (або та і ), написав алгоритми різного рівня складності. Головною проблемою в це раз стало те що я прорахувався з часом для виконання algotester self practice, завдання яке я обрав було складніше ніж я думав, тому воно зайняло більше часу для його виконання.