Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту

A blue and white logo

Description automatically generated

**Звіт**

**Звіт**

**про виконання лабораторних та практичних робіт блоку № 4**

На тему: «Програмування: алгоритм, програма, код. Системи числення. Двійкова система числення. Розробка та середовище розробки програми.»

***з дисципліни:*** «Основи програмування»

до:

Практичних Робіт до блоку № 4

**Виконав:**

Студент групи ШІ-11

Силіч Богдан

Львів 2024

**Тема:** Одновимірні масиви. Двовимірні Масиви. Вказівники та Посилання. Динамічні масиви. Структури даних. Вкладені структури. Алгоритми обробки та робота з масивами та структурами.

**Мета:** зрозуміти основи роботи з одновимірними та двовимірними масивами, а також освоїти концепції вказівників, динамічних масивів і структур даних, що дозволить ефективно реалізовувати алгоритми обробки даних та структури, щоб спростити вирішення складних завдань.

**Теоретичні відомості:**

1. Класи пам'яті у C++

○  Статична пам’ять.

○  Динамічна пам’ять.

○  Поняття стеку.

○  Виділення та вивільнення пам’яті.

1. Вступ до Масивів і Вказівників:

○      Основи масивів: визначення, важливість, приклади використання.

○      Різниця між статичними та динамічними масивами.

○      Основи вказівників: що це таке, як вони працюють.

○      Взаємозв'язок між масивами та вказівниками.

○      Вступ до посилань: основні концепції та відмінності від вказівників.

1. Одновимірні Масиви:

○      Створення та ініціалізація одновимірних масивів.

○      Основні операції: індексація, присвоєння, читання.

○      Цикли та обхід масивів.

○      Використання функцій для роботи з масивами.

○      Приклади алгоритмів сортування та пошуку.

1. Вказівники та Посилання:

○      Використання вказівників для доступу до елементів масиву.

○      Арифметика вказівників.

○      Різниця між вказівниками та посиланнями в контексті функцій.

○      Динамічне виділення пам'яті з використанням вказівників.

○      Використання вказівників для створення складних структур даних.

1. Двовимірні Масиви:

○      Оголошення та ініціалізація двовимірних масивів.

○      Вкладені цикли для обходу двовимірних масивів.

○      Практичні приклади використання двовимірних масивів.

○      Передача двовимірних масивів у функції.

○      Застосування двовимірних масивів для розв'язання задач.

1. Динамічні Масиви:

○      Основи динамічного виділення пам'яті.

○      Створення та управління динамічними масивами.

○      Використання операторів new та delete для управління пам'яттю.

○      Реалізація змінної розмірності масивів.

○      Передача динамічних масивів у функції.

1. Структури Даних:

○      Оголошення та використання структур.

○      Використання масивів та вказівників у структурах.

○      Функції для обробки даних у структурах.

**Індивідуальний план опрацювання теорії:**

Класи пам'яті у C++

Вступ до Масивів і Вказівників

Одновимірні Масиви

Вказівники та Посилання

Двовимірні Масиви

Динамічні Масиви

Структури Даних

1. **Джерела**:

* Chat gpt
* Список відтворення на YouTube ( <https://youtube.com/playlist?list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&si=sXvmPdnGkwvJLXUi> )
* Власний досвід
* Лекції та практичні

**Виконання роботи:**

**VNS Lab 4 Task 1 variant 6**

1) Сформувати одновимірний масив цілих чисел, використовуючи генератор

випадкових чисел.

2) Роздрукувати отриманий масив.

3) Знищити елемент із заданим номером.

4) Додати після першого парного елемента масиву елемент зі значенням M[ I-1

]+2.

5) Роздрукувати отриманий масив.  
**VNS Lab 5 Task 1 variant 6**

Елемент матриці є сідловою точкою, якщо він є найменшим у своєму рядку й

найбільшим у своєму стовпці (або навпаки: найбільшим у своєму рядку й

найменшим у своєму стовпці). Для заданої матриці визначити всі сідлові

точки.

**Algotester Lab 2 variant 3**

Вам дано масив цілих чисел розміром NN, на першій та останній клітинці розміщено по дрону.

Вони одночасно взлітають.

На початку кожного ходу швидкість дрону стає рівною значенню клітинки, у якій він знаходиться.

Тобто лівий дрон у першу секунду з клітинки з індексом 11 перелетить у клітинку з індексом a1a1, тобто його наступна позиція рахується як поточна позиція + число у поточній позиції (перегляньте пояснення для візуалізації) Правий робить аналогічно в протилежну сторону.

Вони це роблять до моменту, коли трапиться одна з зазначених подій:

Якшо 2 дрони опиняються в одній клітинці - ви виводите **Collision**.

Якщо лівий дрон опиниться справа від правого - це **Miss**

У випадку якщо вони зупиняться один навпроти одного, тобто у клітинках ai та ai+1 - виведіть **Stopped**

Врахуйте, що перевіряти треба також до взльоту.

**Algotester Lab 3 variant 6**

Вам дана стрічка s.

Ваше завдання зробити компресію стрічки, тобто якщо якась буква йде більше одного разу підряд у стрічці замінити її на букву + кількість входжень підряд.

**Class Practice Task**

Реалізувати програму, яка перевіряє, чи дане слово чи число є паліндромом за допомогою рекурсії.

Паліндром — це послідовність символів, яка однаково читається вперед і назад (наприклад, «radar», «level», «12321»).

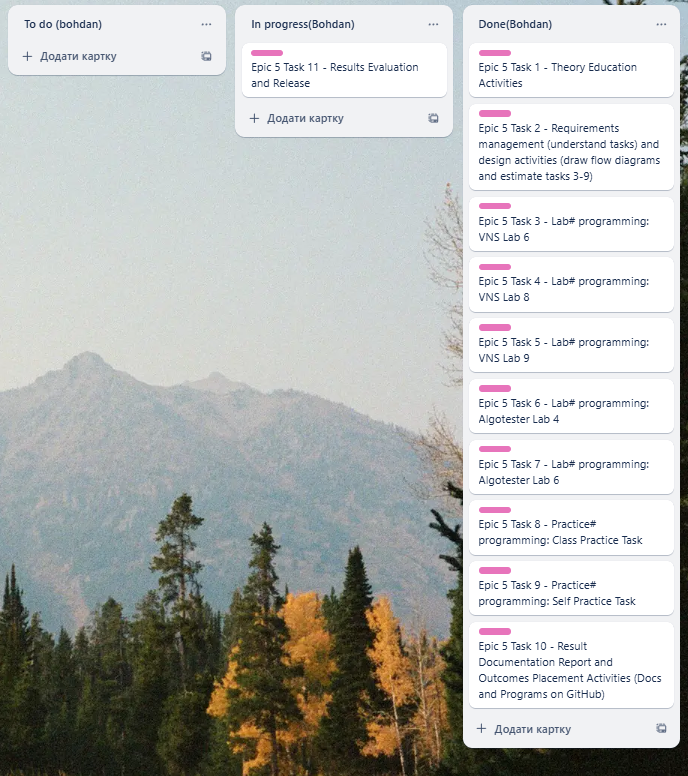
**Self Practice Task**

Зеник і Марічка приїхали до Антарктиди подивитися на пінгвінів. Мабуть, не знали, що у Львові вони теж є.

Усього є n пінгвінів, причому i-ий на початку розміщений у точці (xi,yi) на площині. Ціль пінгвінів — зустрітись разом у певній точці. На жаль, пінгвіни не дуже класно вміють ходити, тому за одну секунду з точки (x,y) пінгвін може перейти хіба що в (x+1,y) або (x,y+1). Ба більше, у будь-який момент часу рухатися може лише один пінгвін.

За який мінімальний час всі пінгвіни можуть зустрітися в певній точці?

1. Requirements management and design activities   
Team Trello dashboard



Team meeting



**UML-diagram block-scheme**

VNS Lab 4 Task 1

35хв+-

VNS Lab 5 Task 1

20хв +-

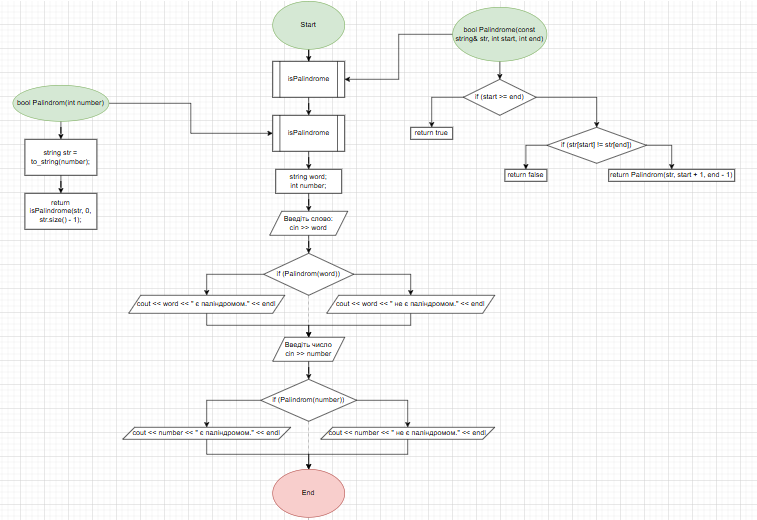
Algotester Lab 2 v3

35хв+-

ALgotester lab 3 v3

45хв+-

Class Practice Task

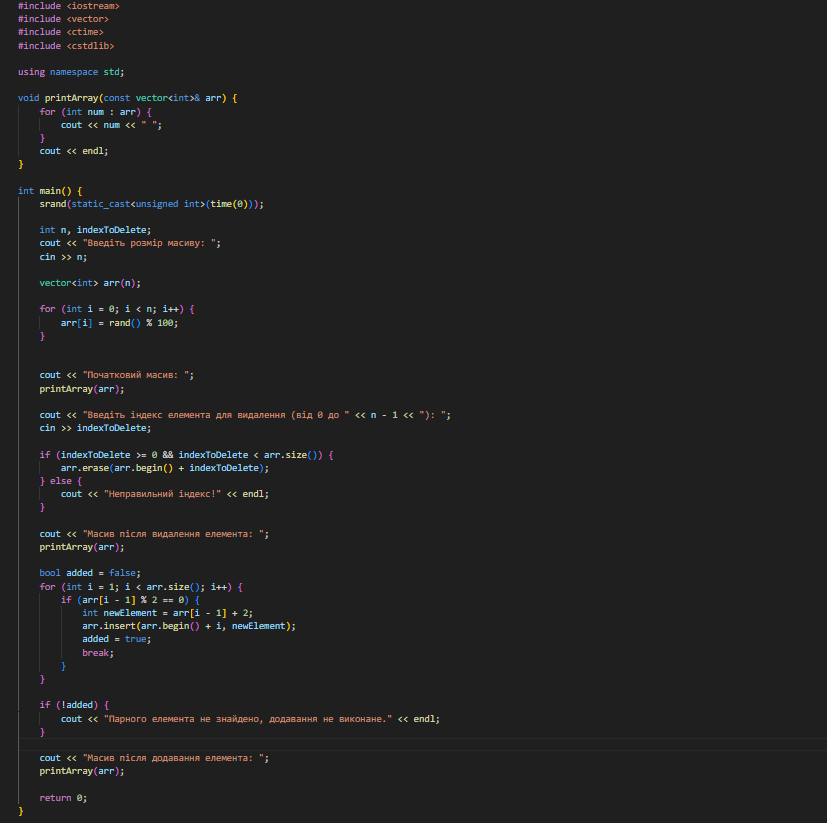
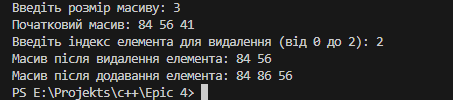


20хв+-

Self Practice Task

30хв+-

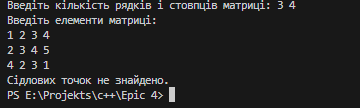
Код програми  
VNS Lab 4 Task 1

+- 30 хв

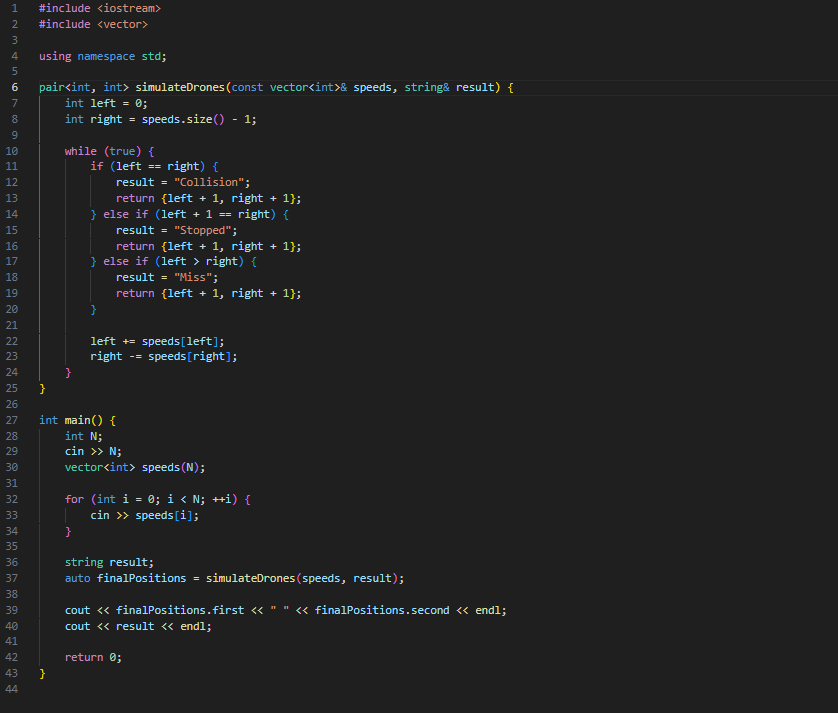
VNS Lab 5 Task 1





+-45 хв

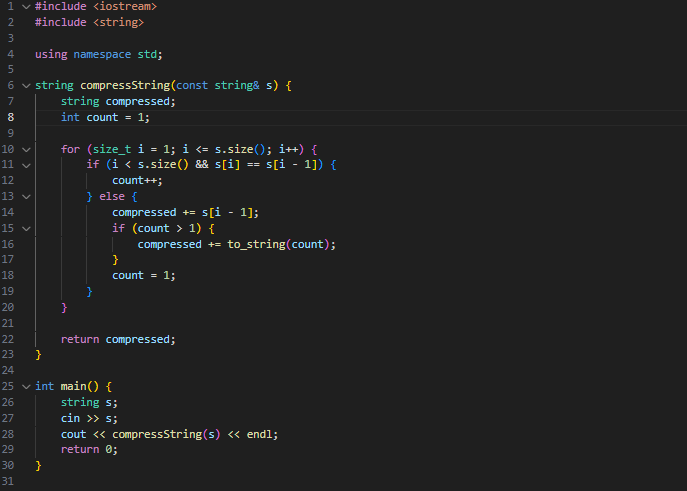
Algotester Lab 2 v3





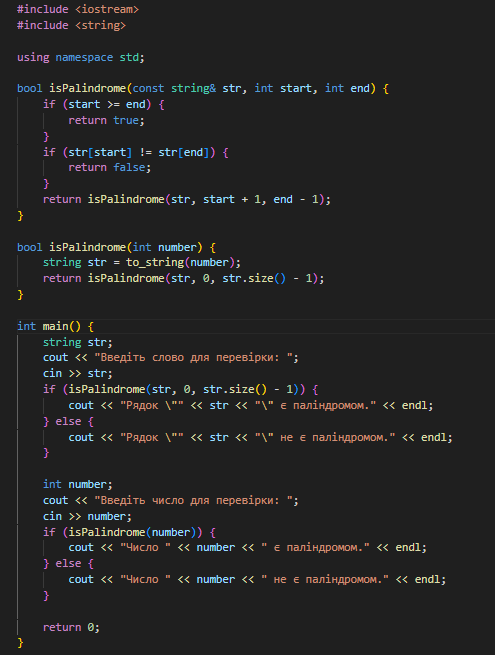
+- 20 хв

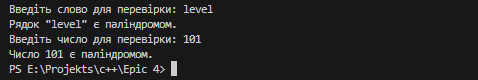
Algotester Lab 3 v3

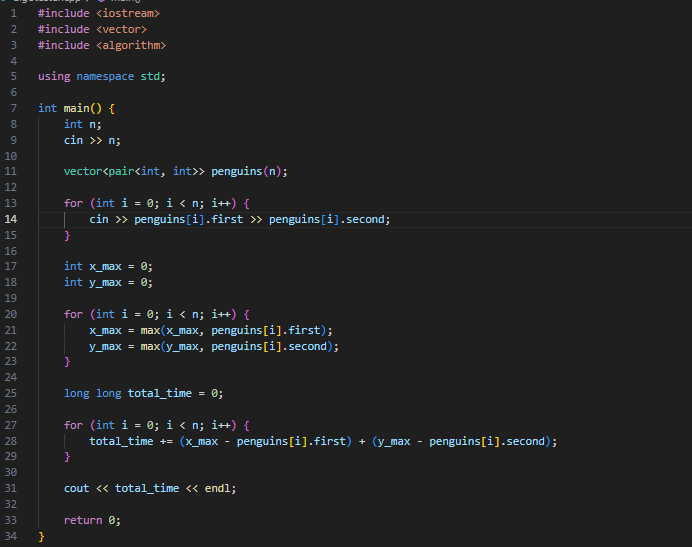
 

25 хв +-

Class Practice Task



  
+- 40 хв

Self Practice Task  


  
+-15 хв

**Висновки:**

Виконуючи цей епік я поглибив свої знання у зберіганні та обробці даних, що суттєво спрощує програмування. Застосуванн вказівників і динамічних масивів дозволяє зекономити пам'ять і гнучко управляти даними, а розуміння структур даних та алгоритмів обробки допомагає створювати оптимізовані рішення для різноманітних задач.