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 № 1-6**

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

**Виконав(-ла):**

Студент групи ШІ-12

Абрамович Дмитро Андрійович

# **Тема роботи:**

*Епік 1*: Розробка, програмування та код. Середовища для розробки.

*Епік 2*:Лінійні алгоритми. Розгалужені алгоритми. Умовні та логічні оператори. Системи числення. Змінні. Константи. Типи даних. Розмір Типів Даних (Двійкова система). Ввід вивід. Базові операції та вбудовані функції. Коментарі.

*Епік 3:* Цикли. Вкладені Цикли. Завершення виконання циклів. Функції. Простір імен. Перевантаження функцій. Функції з змінною кількістю параметрів (еліпсис). Рекурсія. Вбудовані функції.

*Епік 4*: Одновимірні масиви. Двовимірні Масиви. Вказівники та Посилання. Динамічні масиви. Структури даних. Вкладені структури. Алгоритми обробки та робота з масивами та структурами.

*Епік 5*: Файли. Бінарні Файли. Символи і Рядкові Змінні та Текстові Файли. Стандартна бібліотека та деталі/методи роботи з файлами. Створення й використання бібліотек.

*Епік 6*: Динамічні структури (Черга, Стек, Списки, Дерево). Алгоритми обробки динамічних структур.

# **Мета роботи:**

*Епік 1*: Вивчити основи розробки програмного забезпечення, ознайомитися з різними середовищами для розробки, розуміти процес програмування та написання коду.

*Епік 2*: Оволодіти концепціями розгалужених алгоритмів та циклів, вивчити основні операції та структури даних, розрізняти різні системи числення, ознайомитися з роботою змінних та типів даних.

*Епік 3*: Розширити розуміння структур програм, вивчити роботу циклів та функцій, ознайомитися з поняттями простору імен та перевантаження функцій, оволодіти рекурсією та використанням вбудованих функцій. Оволодіти навичками написання функцій з різною кількістю параметрів.

*Епік 4*: Вивчити роботу з масивами та структурами даних, розібратися в використанні вказівників та посилань, освоїти динамічні масиви та алгоритми обробки даних.

*Епік 5*: Оволодіти навичками роботи з файлами, розібратися в особливостях текстових та бінарних файлів, вивчити стандартну бібліотеку та використання бібліотек у програмуванні.

*Епік 6*: Оволодіти роботою з динамічними структурами даних, такими як черга, стек, списки та дерева, вивчити алгоритми обробки цих структур для ефективного програмування.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* *Епік 1:*
* Тема №1: Trello
* Тема №2: Git and GitHub
* Тема №3: Flowcharts and Draw.io
* Тема №4: Visual Studio Code
* Тема №5: С++ Basics
* *Епік 2:*
* Тема №1: Константи та змінні
* Тема №2: Умовні оператори if-else
* Тема №3: Оператор switch-case
* Тема №4: Оператори циклів
* *Епік 3:*
* Тема №1: Функції
* Тема №2: Перевантаження функцій
* Тема №3: Рекурсія
* *Епік 4:*
* Тема №1: Прості структури даних
* Тема №2: Одновимірні масиви
* Тема №3: Двовимірні Масиви
* Тема №4: Алгоритми обробки
* *Епік 5:*
* Тема №1: Потоковий ввід-вивід
* Тема №2: Бінарні файли
* Тема №3: Робота з файлами
* *Епік 6:*
* Тема №1: Черга
* Тема №2: Однозв’язний список
* Тема №3: Двозв’язний список
* Тема №4: Бінарне дерево
* Тема №5: Бінарне дерево пошуку

1. Індивідуальний план опрацювання теорії:
   1. *Епік 1:*

* *Тема №1*: Trello.
  + Джерела Інформації
    - <https://trello.com/guide>
    - <https://www.youtube.com/watch?v=lfMJ-NeoUGM>
    - <https://www.youtube.com/watch?v=7dIuGc6SxuA>
    - <https://www.cloudwards.net/trello-beginners-guide/>
    - <https://www.simplilearn.com/tutorials/project-management-tutorial/what-is-trello>
    - <https://blog.trello.com/how-to-use-trello-like-a-pro>
  + Що опрацьовано:
  + Суть роботи з Trello. Призначення платформи та організація роботи на дошках.
  + Статус: Ознайомлений(-а).
* *Тема №2*: Git and Github.
  + Джерела Інформації
    - <https://git-scm.com/download/win>
    - <https://git-scm.com/download/mac>
    - <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>
    - <https://www.msys2.org/docs/git/>
    - [https://www.youtube.com/watch?v=vR-y\_2zWrIE](https://www.youtube.com/watch?v=vR-y_2zWrIE&list=PLWKjhJtqVAbkFiqHnNaxpOPhh9tSWMXIF&ab_channel=freeCodeCamp.org)
    - <https://www.youtube.com/watch?v=RGOj5yH7evk&ab_channel=freeCodeCamp.org>
    - <https://www.youtube.com/watch?v=ySKJF3ewfVk&list=PLJ6ZMUSN40FF8pBX4bv1mhVIgoem33Zfv&ab_channel=programmingmentorua>
    - <https://www.freecodecamp.org/news/introduction-to-git-and-github/>
    - <https://www.wiley.com/en-ie/GitHub+For+Dummies-p-9781119572657>
    - <https://www.msys2.org/docs/package-management/>
    - <https://www.msys2.org/docs/what-is-msys2/>
    - <https://manpages.ubuntu.com/manpages/jammy/man8/apt-get.8.html>
    - <https://docs.brew.sh/>
    - <https://www.freecodecamp.org/news/the-linux-commands-handbook/>
    - <https://docs.github.com/en/get-started>
  + Що опрацьовано:
  + Суть систем контролю версій, базова робота з розподіленою СКН Git, хостинг репозиторію Git на Github.
  + Статус: Ознайомлений(-а).
* *Тема №3*: Flowcharts and Draw.io.
  + Джерела Інформації
    - <https://www.gliffy.com/blog/guide-to-flowchart-symbols>
    - <https://en.wikipedia.org/wiki/Flowchart>
    - <https://www.lucidchart.com/pages/what-is-a-flowchart-tutorial>
  + Що опрацьовано:
  + Призначення блок-схем та різних видів блоків, робота з платформою Draw.io.
  + Статус: Ознайомлений(-а).
* *Тема №4*: Visual Studio Code.
  + Джерела Інформації
    - <https://code.visualstudio.com/docs/cpp/config-mingw>
    - <https://code.visualstudio.com/docs/cpp/config-linux>
    - <https://code.visualstudio.com/docs/cpp/config-clang-mac>
    - <https://www.youtube.com/watch?v=2VokW_Jt0oM&ab_channel=ProgrammingKnowledge>
    - [How to Set up Visual Studio Code for C and C++ Programming](https://www.youtube.com/watch?v=77v-Poud_io&ab_channel=LearningLad)
    - [Debug a C++ project in VS Code](https://www.youtube.com/watch?v=G9gnSGKYIg4&pp=ygUgdmlzdWFsIHN0dWRpbyBjb2RlIGMrKyBkZWJ1Z2dpbmc%3D)
  + Що опрацьовано:
  + Встановлено Visual Studio Code, додано пакет розширень для С++, настроєно MinGW та G++ для роботи з С++.
  + Статус: Ознайомлений(-а).
* *Тема №5*: C++ Basics.
  + Джерела Інформації
    - <https://www.w3schools.com/cpp/>
    - <https://learn.microsoft.com/en-us/cpp/cpp>
    - <https://en.cppreference.com/w/>
    - <https://www.geeksforgeeks.org/cpp-tutorial/>
    - <https://www.learncpp.com/>
    - <https://www.tutorialspoint.com/cplusplus/index.htm>
  + Що опрацьовано:
  + Базові принципи С++, типи даних, консольний ввід та вивід.
  + Статус: Ознайомлений(-а).
  1. *Епік 2:*
* Тема №1: Константи та змінні.
  + Джерела Інформації

Лекції:

<https://vns.lpnu.ua/pluginfile.php?file=%2F1098482%2Fmod_resource%2Fcontent%2F1%2F%D0%9B%D0%B5%D0%BA%D1%86%D1%96%D0%B9%D0%BD%D0%B8%D0%B9%20%D0%BC%D0%B0%D1%82%D0%B5%D1%80%D1%96%D0%B0%D0%BB%20%D0%B4%D0%BE%20%D1%82%D0%B5%D0%BC%D0%B8%20%E2%84%96%202.pdf>

* + Що опрацьовано:
  + Декларування констант, оголошення змінних та принцип роботи з ними
  + Статус: Ознайомлений
* Тема №2: Умовні оператори if-else
  + Джерела Інформації:
    - Стаття:

<https://lib.chmnu.edu.ua/pdf/posibnuku/350/6.pdf>

* + Що опрацьовано:

Опрацьовано матеріал про умовні оператори if-else

* + Статус: Ознайомлений
* Тема №3: Оператор switch-case
  + Джерела Інформації:
    - Стаття:

<https://cherto4ka.xyz/2018/10/20/%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80-%D0%BC%D0%BD%D0%BE%D0%B6%D0%B8%D0%BD%D0%BD%D0%BE%D0%B3%D0%BE-%D0%B2%D0%B8%D0%B1%D0%BE%D1%80%D1%83-switch/>

* + Що опрацьовано:

Опрацьовано матеріал про оператор switch-case

* + Статус: Ознайомлений
* Тема №4 Оператор повторення for
  + Джерела Інформації:
    - Стаття:

<http://cpp.dp.ua/operatory-tsyklu/>

* + Що опрацьовано:

Опрацьовано матеріал про оператор повторення for

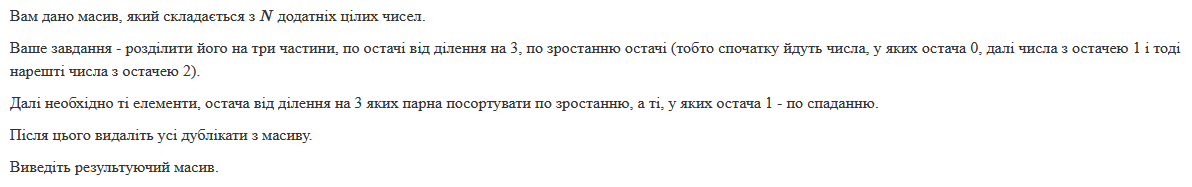
* + Статус: Ознайомлений
  1. *Епік 3:*
* Тема №1: Функції.
  + Джерела Інформації:
    - [Стаття](https://www.w3schools.com/cpp/cpp_functions.asp" \o "https://www.w3schools.com/cpp/cpp_functions.asp)
    - [Відео](https://www.youtube.com/watch?v=HFH65-sNQmI&pp=ygUT0YTRg9C90LrRhtGW0YUg0YErKw%3D%3D" \o "https://www.youtube.com/watch?v=HFH65-sNQmI&pp=ygUT0YTRg9C90LrRhtGW0YUg0YErKw%3D%3D)
    - [Відео№2](https://www.youtube.com/watch?v=V_8XRRIus7Y" \o "https://www.youtube.com/watch?v=V_8XRRIus7Y)
  + Що опрацьовано:
    - Проведено роботу із базовими функціями.
  + Статус: Ознайомлений
* Тема №2: Перевантаження функції.
  + Джерела Інформації:
    - [Відео](https://www.youtube.com/watch?v=3KJfisev6SI&pp=ygUw0L_QtdGA0LXQstCw0L3RgtCw0LbQtdC90L3RjyDRhNGD0L3QutGG0ZbQuSDRgSsr" \o "https://www.youtube.com/watch?v=3KJfisev6SI&pp=ygUw0L_QtdGA0LXQstCw0L3RgtCw0LbQtdC90L3RjyDRhNGD0L3QutGG0ZbQuSDRgSsr)
    - [Відео№2](https://www.youtube.com/watch?v=3GALXU-F3C8&list=PLDyJYA6aTY1llzwya3FrWX4tmo-hm491p&index=11&pp=iAQB" \o "https://www.youtube.com/watch?v=3GALXU-F3C8&list=PLDyJYA6aTY1llzwya3FrWX4tmo-hm491p&index=11&pp=iAQB)
    - [Стаття](https://www.geeksforgeeks.org/function-overloading-c/" \o "https://www.geeksforgeeks.org/function-overloading-c/)
  + Що опрацьовано:
    - Освоєно перевантаження функції та особливості його використання.
  + Статус: Ознайомлений
* Тема №3 Рекурсія.
  + Джерела Інформації:
    - [Відео](https://www.youtube.com/watch?v=_WGNSVDb0t8&pp=ygUWINGBKysg0YDQtdC60YPRgNGB0LjRjw%3D%3D" \o "https://www.youtube.com/watch?v=_WGNSVDb0t8&pp=ygUWINGBKysg0YDQtdC60YPRgNGB0LjRjw%3D%3D)
    - [Стаття](http://cpp.dp.ua/rekursyvni-funktsiyi/" \o "http://cpp.dp.ua/rekursyvni-funktsiyi/)
    - [Стаття№2](https://www.bestprog.net/uk/2019/01/07/recursion-examples-of-tasks-solving-advantages-and-disadvantages-of-recursion-ua-2/" \o "https://www.bestprog.net/uk/2019/01/07/recursion-examples-of-tasks-solving-advantages-and-disadvantages-of-recursion-ua-2/)
  + Що опрацьовано:
    - Освоєно рекурсію та випадки її використання.
  + Статус: Ознайомлений
  1. *Епік 4:*
* Тема №1: Прості структури даних.
  + Джерела Інформації
    - [Data Structures Explained for Beginners - How I Wish I was Taught - YouTube](https://www.youtube.com/watch?v=SFEROgwxicA)
  + Що опрацьовано:
    - Вивчив прості структури даних по туторіалу в ютубі
  + Статус: Ознайомлено
* Тема №2: Одновимірні масиви
  + Джерела Інформації:
    - [C++ Теорія ⦁ Урок 40 ⦁ Одновимірні масиви - YouTube](https://www.youtube.com/watch?v=ULdbOaMBPYc)
  + Що опрацьовано:
    - Вивчив одновимірні масиви та їх використання по туторіалу на ютубі
  + Статус: Ознайомлено
* Тема №3: Двовимірні Масиви
  + Джерела Інформації:
    - [Двовимірні масиви С++ (youtube.com)](https://www.youtube.com/watch?v=6dagAMyWTdY)
  + Що опрацьовано:
    - Ознайомився з двовимірними масивами по роліку на ютубі.
  + Статус: Ознайомлено
* Тема №4: Алгоритми обробки.
  + Джерела Інформації:
    - [C++ Теорія ⦁ Урок 175 ⦁ STL ⦁ Алгоритми - YouTube](https://www.youtube.com/watch?v=m9ZHTiW-P7I)
  + Що опрацьовано:
    - Вивчив застосування алгоритмів обробки по відеоролику на ютубі
  + Статус: Ознайомлено
* Тема №5: Сортування вибіркою
  + Джерела Інформації:
    - [C++ Теорія ⦁ Урок 51 ⦁ Сортування вибіркою - YouTube](https://www.youtube.com/watch?v=uQxG9gBROog)
  + Що опрацьовано:
    - Вивчив застосування сортування вибіркою в коді по відеоролику на ютубі
  + Статус: Ознайомлено
  1. *Епік 5:*
* Тема №1: Потоковий ввід-вивід.
  + Джерела Інформації
    - <http://cpp.dp.ua/potokove-vvedennya-vyvedennya/>
    - <https://acode.com.ua/urok-215-potoky-vvodu-i-vyvodu/>
  + Що опрацьовано:
    - Ввід/вивід в C++
    - Стандартні потоки С++
  + Статус: Ознайомлений
* Тема №2: Блоковий ввід-вивід
  + Джерела Інформації:
  + https://purecodecpp.com/uk/archives/2751Що опрацьовано:
    - Функції, що використовуються для блочного вводу та виводу
  + Статус: Ознайомлений
* Тема №3: Робота з файлами
  + Джерела Інформації:
    - http://www.kytok.org.ua/post/vector-u-cplusplus
    - https://www.bestprog.net/uk/2021/10/12/c-the-vector-class-methods-that-define-and-modify-the-general-characteristics-of-array-ua/
    - https://acode.com.ua/urok-99-vvedennya-v-std-vector/
  + Статус: Ознайомлений
  1. *Епік 6:*
* Тема №1: Черга.
  + Джерела Інформації
    - <https://www.geeksforgeeks.org/introduction-to-queue-data-structure-and-algorithm-tutorials/> .
  + Що опрацьовано:
    - Основний принцип роботи черги, способи її імплементації, функціонал.
  + Статус: Ознайомлений
* Тема №2: Однозв’язний список.
  + - <https://www.scaler.com/topics/linked-list-in-cpp/>
    - <https://youtu.be/m7rrk65GiXY>
  + Що опрацьовано:
    - Як влаштовані однозв’язні списки, як їх створювати та обробляти: функції для роботи з такими списками
  + Статус: Ознайомлений
* Тема №3: Двозв’язний список.
  + Джерела Інформації
    - <https://www.scaler.com/topics/doubly-linked-list-cpp/>
  + Що опрацьовано:
    - Структура елементів двозв’язного списку, відмінність одно- та двозв’язного списку
  + Статус: Ознайомлений
* Тема №4 Бінарне дерево.
  + - <https://www.youtube.com/watch?v=_IhTp8q0Mm0> .
    - <https://www.geeksforgeeks.org/introduction-to-tree-data-structure-and-algorithm-tutorials/?ref=shm>
  + Що опрацьовано:
    - Структура бінарного дерева, основні принципи роботи з деревом
  + Статус: Ознайомлений
* Тема №5: Бінарне дерево пошуку.
  + Джерела Інформації
    - <https://www.geeksforgeeks.org/introduction-to-binary-search-tree-data-structure-and-algorithm-tutorials/>
  + Що опрацьовано:
    - Використання бінарних дерев пошуку, відмінність від звичайних бінарних дерев
  + Статус: Ознайомлений

# **Виконання роботи:**

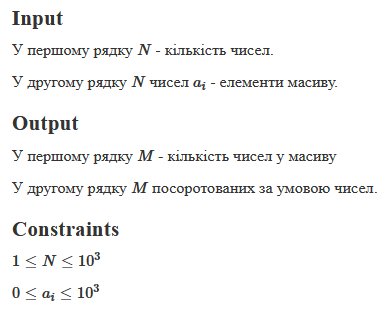
## **1. Опрацювання завдання та вимог до програм та середовища:**

Завдання №1 Algotester Lab 1

* Варіант завдання - 1
* Деталі завдання

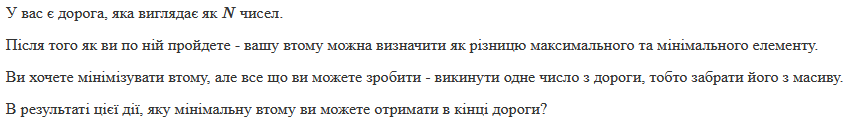


* Важливі деталі для врахування в імплементації програми

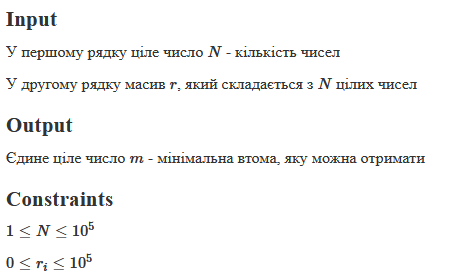


Завдання №2 Algotester Lab 2

* Варіант завдання - 1
* Деталі завдання

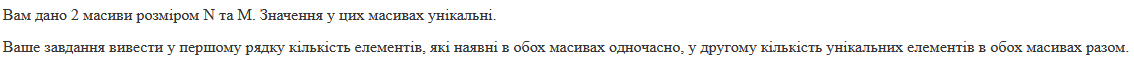


* Важливі деталі для врахування в імплементації програми

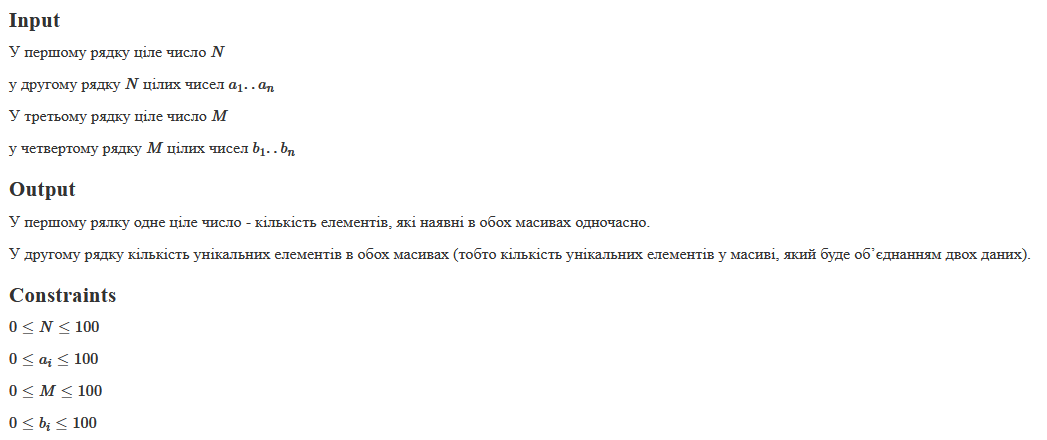


Завдання №3 Algotester Lab 3

* Варіант завдання - 2
* Деталі завдання

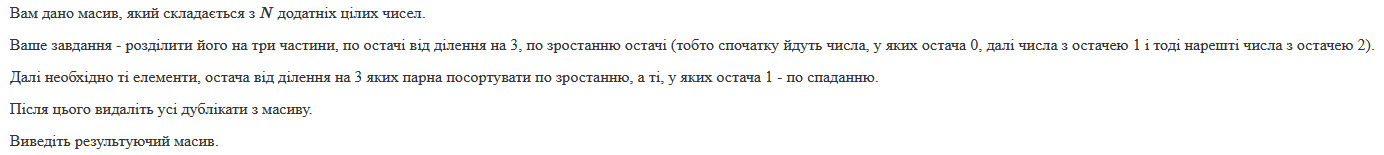


* Важливі деталі для врахування в імплементації програми

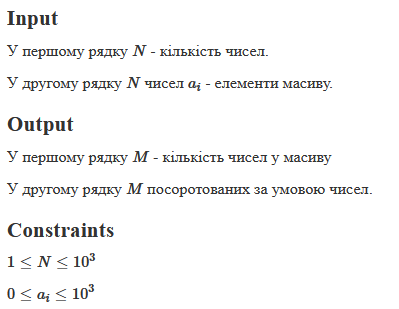


Завдання №4 Algotester Lab 4

* Варіант завдання - 3
* Деталі завдання



* Важливі деталі для врахування в імплементації програми

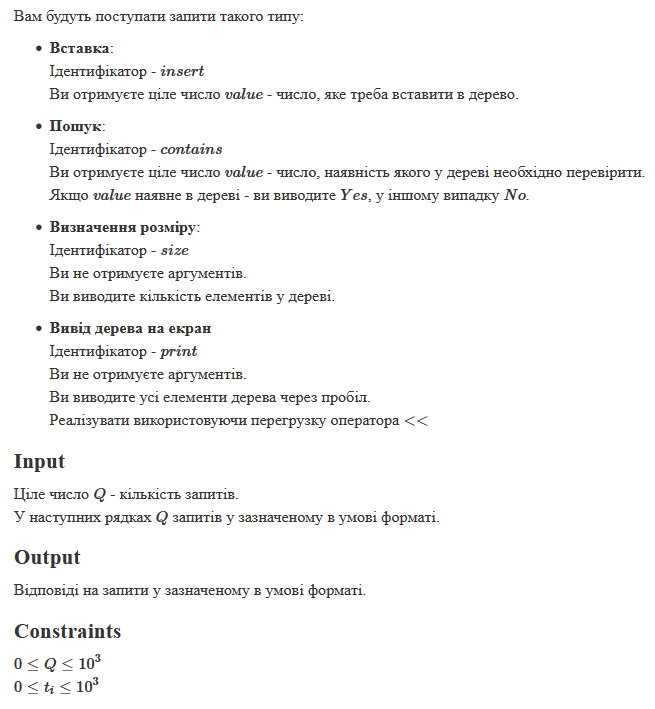


Завдання №5 Algotester Lab 7-8

* Варіант завдання - 3
* Деталі завдання



* Важливі деталі для врахування в імплементації програми



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

Програма №1 Algotester Lab 1 <ПРИКЛАД>

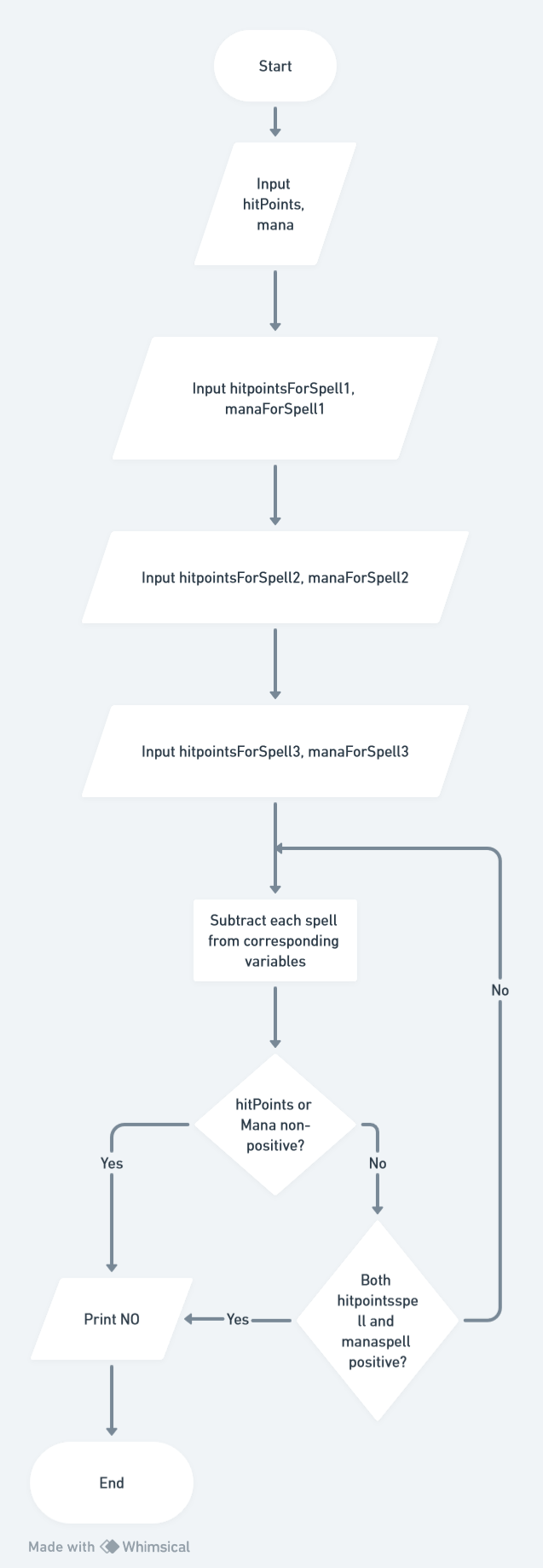


Рисунок 1: Блок-схема до програми №1

* Планований час на реалізацію - 1 година.

Програма №2 Algotester Lab 2

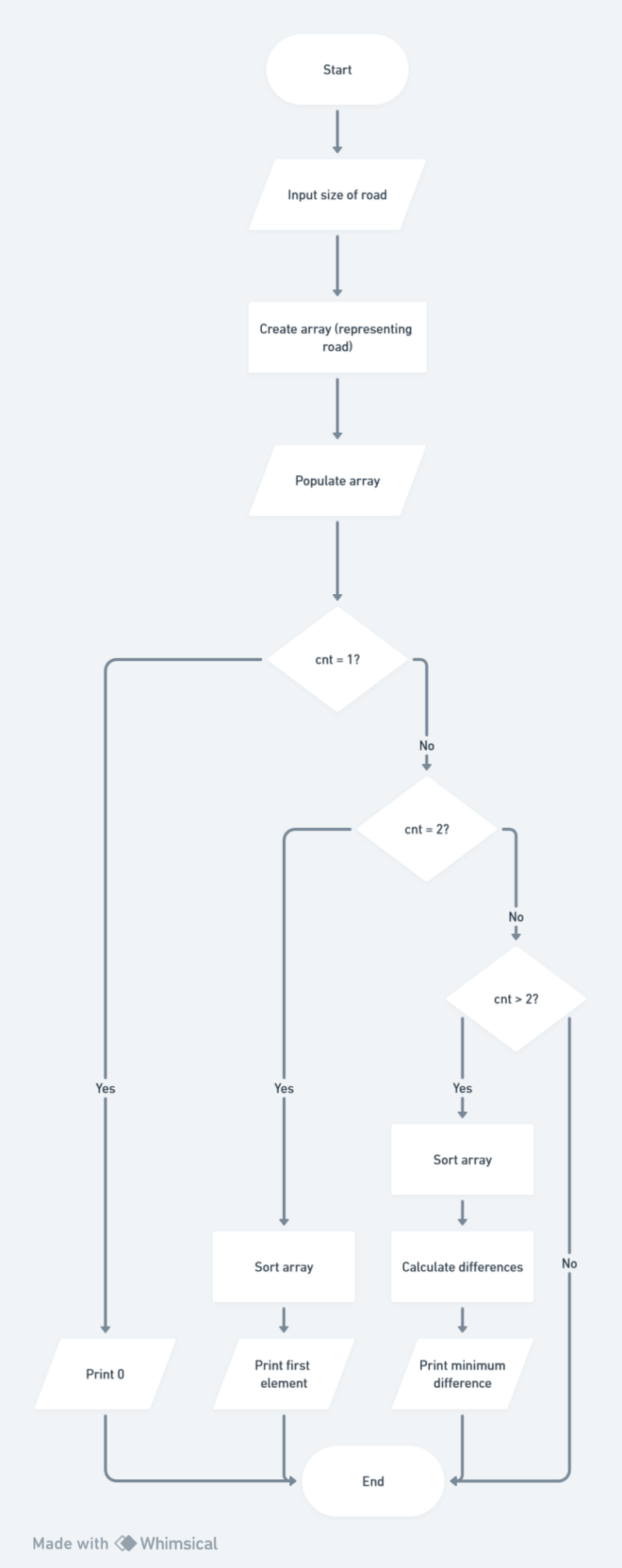


Рисунок 2: Блок-схема до програми №2

* Планований час на реалізацію - 1 година.

Програма №3 Algotester Lab 3

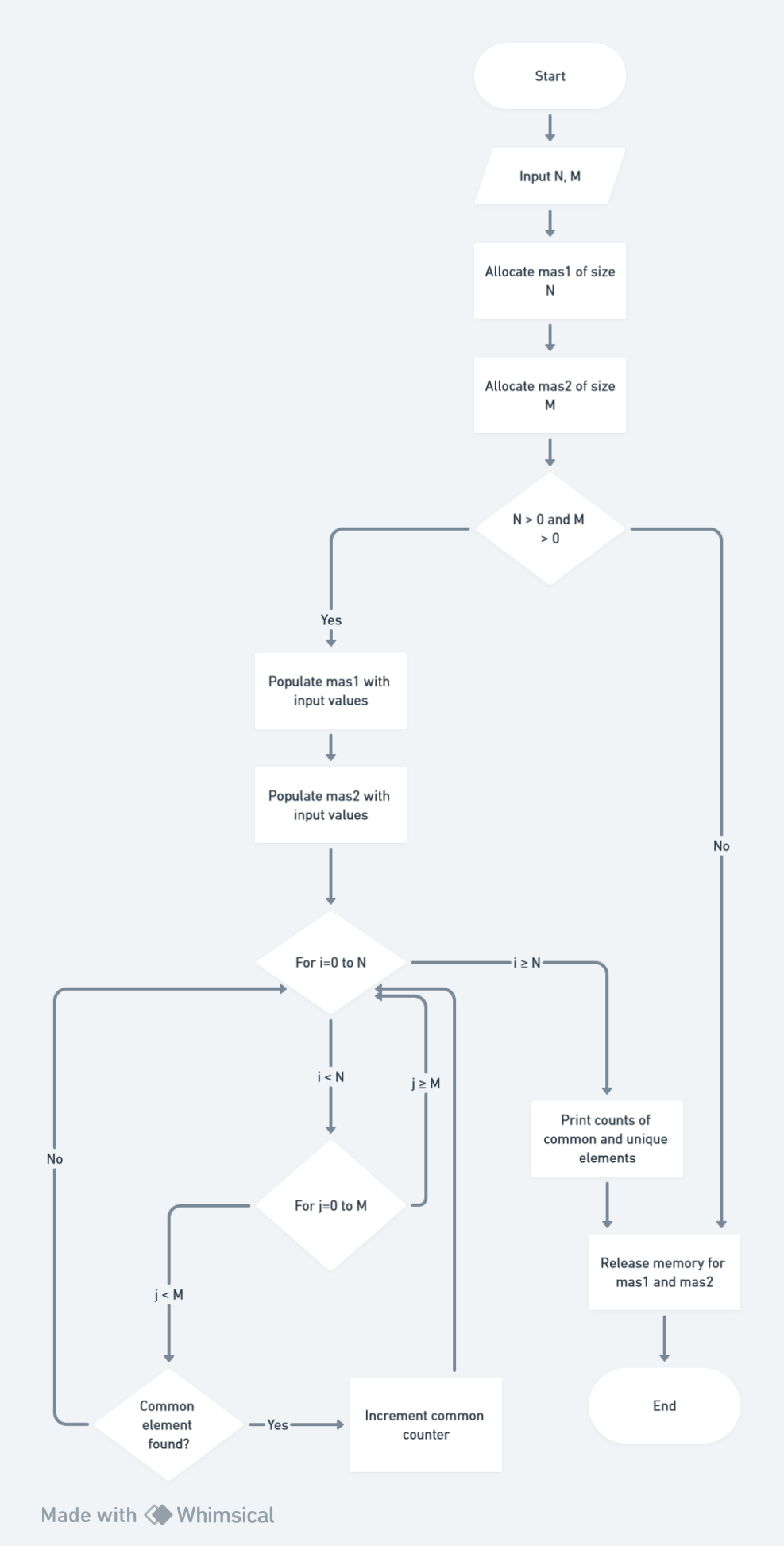


Рисунок 3: Блок-схема до програми №3

* Планований час на реалізацію - 1 година.

Програма №4 Algotester Lab 4

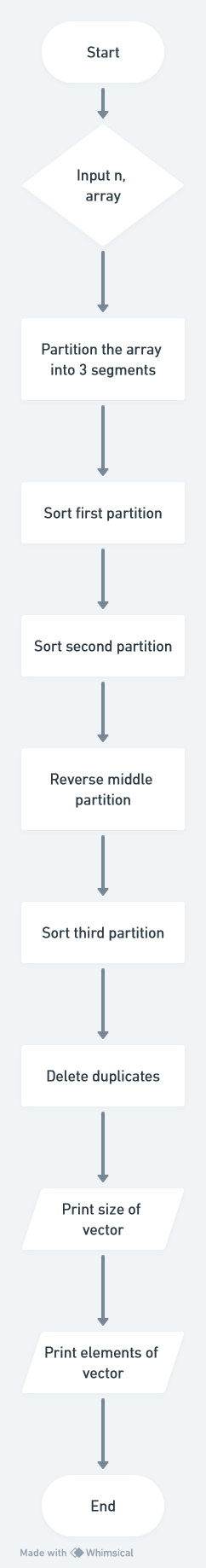


Рисунок 4: Блок-схема до програми №4

* Планований час на реалізацію - 1 година.

Програма №5 Algotester Lab 7-8

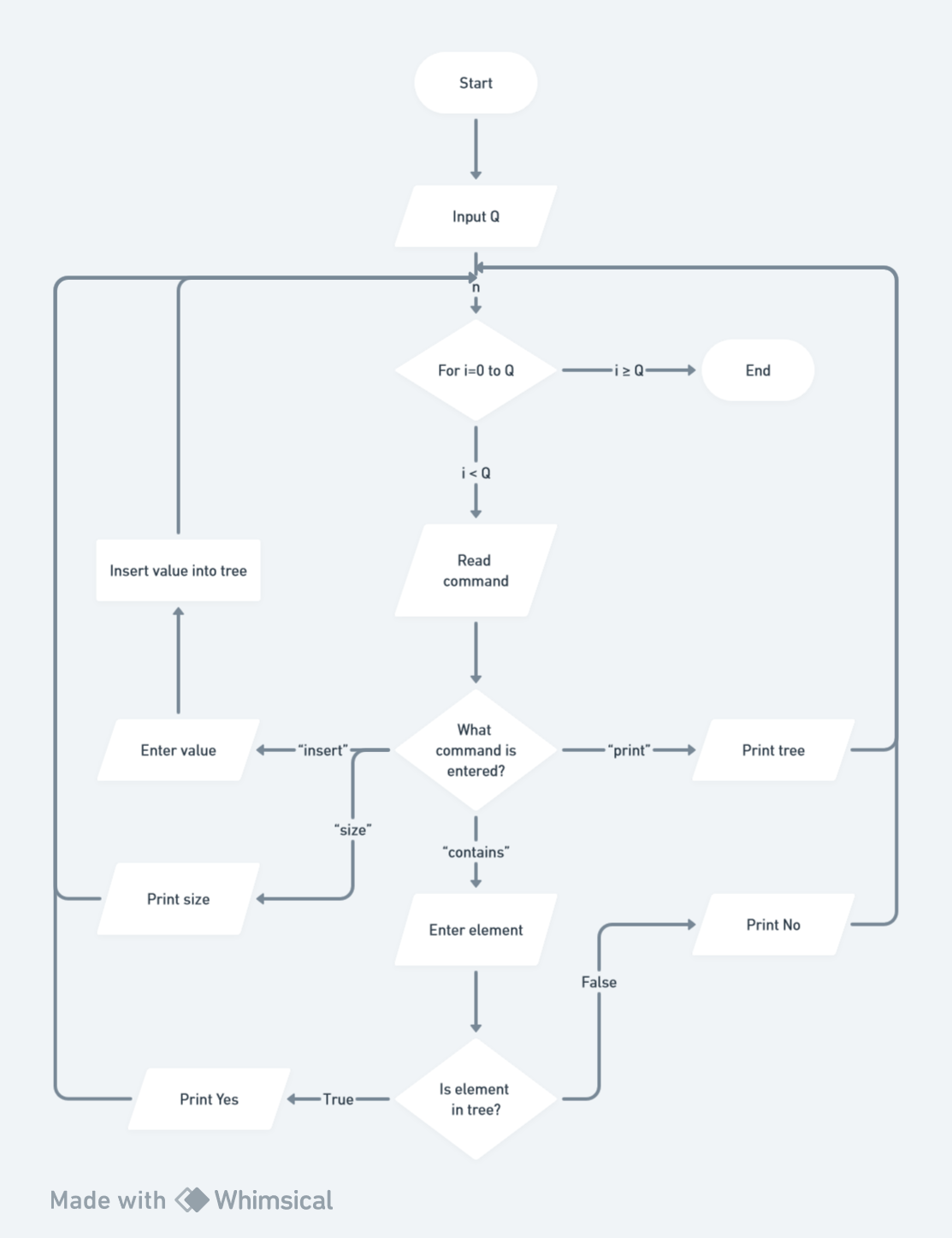


Рисунок 5: Блок-схема до програми №5

* Планований час на реалізацію - 1 день.

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

Завдання №1 Algotester Lab 1

#include <iostream>

using namespace std;

int main() {

    long long   hitPoints, mana,

                hitpointsForSpell, manaForSpell;

    cin >> hitPoints >> mana;

    for (int i = 0; i < 3; ++i) {

        cin >> hitpointsForSpell >> manaForSpell;

        hitPoints -= hitpointsForSpell;

        mana -= manaForSpell;

        if ((hitPoints <= 0 || mana <= 0) ||

            (hitpointsForSpell > 0 && manaForSpell > 0)) {

            printf("NO\n");

            return 0;

        }

    }

    printf("YES\n");

    return 0;

}

Завдання №2 Algotester Lab 2

#include <iostream>

#include <algorithm>

using namespace std;

int main()

{

    unsigned int cnt;

    cin >> cnt;

    int \*arr = new int[cnt];

    for (unsigned int i = 0; i < cnt; i++)

    {

        cin >> arr[i];

    }

    sort(arr, arr + cnt);

    switch (cnt)

    {

    case 1:

        cout << 0;

        break;

    case 2:

        cout << arr[0];

        break;

    default:

        if (arr[cnt - 1] - arr[1] > arr[cnt - 2] - arr[0])

            cout << arr[cnt - 2] - arr[0];

        else

            cout << arr[cnt - 1] - arr[1];

    }

    return 0;

}

Завдання №3 Algotester Lab 3

#include <iostream>

using namespace std;

int main() {

    int N, M, unique, common = 0;

    int\* mas1, \*mas2;

    cin >> N;

    mas1 = new int[N];

    for (int i = 0; i < N; ++i) cin >> mas1[i];

    cin >> M;

    mas2 = new int[M];

    for (int i = 0; i < M; ++i) cin >> mas2[i];

    for (int ind1 = 0; ind1 < N; ++ind1)

        for (int ind2 = 0; ind2 < M; ++ind2)

            if (mas1[ind1] == mas2[ind2]) {

                common++;

                break;

            }

    unique = N + M - common;

    cout << common << '\n' << unique << '\n';

    delete[] mas1;

    delete[] mas2;

    return 0;

}

Завдання №4 Algotester Lab 4

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

void merge(vector<int>& arr, int left, int mid, int right) {

    int n1 = mid - left + 1;

    int n2 = right - mid;

    vector<int> leftArr(n1);

    vector<int> rightArr(n2);

    for (int i = 0; i < n1; i++) {

        leftArr[i] = arr[left + i];

    }

    for (int i = 0; i < n2; i++) {

        rightArr[i] = arr[mid + 1 + i];

    }

    int i = 0, j = 0, k = left;

    while (i < n1 && j < n2) {

        if (leftArr[i] <= rightArr[j]) {

            arr[k] = leftArr[i];

            i++;

        } else {

            arr[k] = rightArr[j];

            j++;

        }

        k++;

    }

    while (i < n1) {

        arr[k] = leftArr[i];

        i++;

        k++;

    }

    while (j < n2) {

        arr[k] = rightArr[j];

        j++;

        k++;

    }

}

void merge\_sort(vector<int>& arr, int left, int right) {

    if (left < right) {

        int mid = left + (right - left) / 2;

        merge\_sort(arr, left, mid);

        merge\_sort(arr, mid + 1, right);

        merge(arr, left, mid, right);

    }

}

void reverse(vector<int>::iterator start, vector<int>::iterator end) {

    while (start < end) {

        swap(\*start, \*end);

        start++;

        end--;

    }

}

void mergeSort(vector<int>& arr) {

    merge\_sort(arr, 0, arr.size() - 1);

}

int main() {

    int n;

    cin >> n;

    vector<int> vec(n);

    for (int i = 0; i < n; ++i) cin >> vec[i];

    auto part1 = partition(vec.begin(), vec.end(), [](int a) { return a % 3 == 0; });

    auto part2 = partition(part1, vec.end(), [](int a) { return a % 3 == 1; });

    merge\_sort(vec, 0, distance(vec.begin(), part1) - 1);

    merge\_sort(vec, distance(vec.begin(), part1), distance(vec.begin(), part2) - 1);

    reverse(part1, part2 - 1);

    merge\_sort(vec, distance(vec.begin(), part2), n - 1);

    int m = 0;

    for (int i = 0; i < n; i++)

        if (i == 0 || vec[i] != vec[i - 1]) {

            vec[m] = vec[i];

            m++;

        }

    cout << m << endl;

    for (int i = 0; i < m; ++i)

        cout << vec[i] << " ";

    return 0;

}

Завдання №5 Algotester Lab 7-8

#include <iostream>

#include <functional>

using namespace std;

template <typename T>

class BinaryTree

{

private:

    struct TreeNode

    {

    public:

        T value;

        TreeNode \*parent;

        TreeNode \*left;

        TreeNode \*right;

        TreeNode(T value, TreeNode \*parent, TreeNode \*left, TreeNode \*right)

            : value(value), parent(parent), left(left), right(right) {}

        TreeNode(T value)

            : TreeNode(value, NULL, NULL, NULL) {}

    };

    void put\_node(TreeNode \*inserted\_node, TreeNode \*\*head)

    {

        if (inserted\_node == NULL)

            return;

        if (\*head == NULL)

        {

            \*head = inserted\_node;

            tree\_size++;

            return;

        }

        if (inserted\_node->value < (\*head)->value)

        {

            if ((\*head)->left == NULL)

            {

                (\*head)->left = inserted\_node;

                inserted\_node->parent = (\*head);

                tree\_size++;

            }

            else

                put\_node(inserted\_node, &(\*head)->left);

        }

        else if (inserted\_node->value > (\*head)->value)

        {

            if ((\*head)->right == NULL)

            {

                (\*head)->right = inserted\_node;

                inserted\_node->parent = (\*head);

                tree\_size++;

            }

            else

                put\_node(inserted\_node, &(\*head)->right);

        }

    }

    void destruct\_tree(TreeNode \*node)

    {

        if (node == NULL)

            return;

        destruct\_tree(node->left);

        destruct\_tree(node->right);

        delete node;

    }

    TreeNode \*lookup\_for\_node(T \_val)

    {

        TreeNode \*pNode = tree\_head;

        while (pNode != NULL)

        {

            if (pNode->value == \_val)

                return pNode;

            if (\_val < pNode->value)

                pNode = pNode->left;

            else

                pNode = pNode->right;

        }

        return NULL;

    }

    void for\_each(TreeNode \*head, function<void(T)> delegate)

    {

        if (head == NULL)

            return;

        for\_each(head->left, delegate);

        delegate(head->value);

        for\_each(head->right, delegate);

    }

    TreeNode \*tree\_head;

    int tree\_size;

public:

    BinaryTree()

    {

        tree\_head = NULL;

        tree\_size = 0;

    }

    ~BinaryTree()

    {

        destruct\_tree(tree\_head);

    }

    int size()

    {

        return tree\_size;

    }

    bool contains\_value(T value)

    {

        return lookup\_for\_node(value) != NULL;

    }

    void append(T value)

    {

        put\_node(new TreeNode(value), &tree\_head);

    }

    void for\_each(function<void(T)> func)

    {

        for\_each(tree\_head, func);

    }

    void print\_to\_stream(ostream &os)

    {

        for\_each([&os](T value)

                 { os << value << " "; });

    }

    void print()

    {

        print\_to\_stream(cout);

    }

    friend ostream &operator<<(ostream &os, BinaryTree<T> &\_tree)

    {

        \_tree.print\_to\_stream(os);

        return os;

    }

};

int main()

{

    BinaryTree<int> tree;

    int intbuf, n;

    string strbuf;

    cin >> n;

    for (int i = 0; i < n; i++)

    {

        cin >> strbuf;

        if (strbuf == "insert")

        {

            cin >> intbuf;

            tree.append(intbuf);

        }

        else if (strbuf == "contains")

        {

            cin >> intbuf;

            cout << (tree.contains\_value(intbuf) ? "Yes" : "No") << endl;

        }

        else if (strbuf == "size")

        {

            cout << tree.size() << endl;

        }

        else if (strbuf == "print")

        {

            cout << tree << endl;

        }

    }

    return 0;

}

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

Завдання №1 Algotester Lab 1

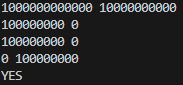


Рисунок 6: Приклад виконання програми №1

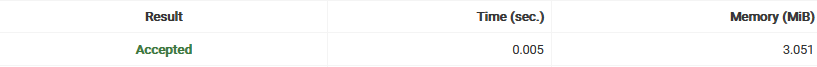


Рисунок 7: Факт прийняття розв’язку задачі середовищем Algotester

Час затрачений на виконання завдання - 2 години.

Завдання №2 Algotester Lab 2



Рисунок 8: Приклад виконання програми №1



Рисунок 9: Факт прийняття розв’язку задачі середовищем Algotester

Час затрачений на виконання завдання - 2 години.

Завдання №3 Algotester Lab 3

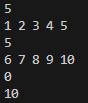


Рисунок 10: Приклад виконання програми №1

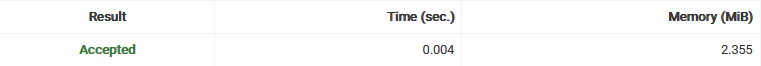


Рисунок 11: Факт прийняття розв’язку задачі середовищем Algotester

Час затрачений на виконання завдання - 3 години.

Завдання №4 Algotester Lab 4

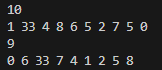


Рисунок 12: Приклад виконання програми №1



Рисунок 13: Факт прийняття розв’язку задачі середовищем Algotester

Час затрачений на виконання завдання - 3 години.

Завдання №5 Algotester Lab 7-8

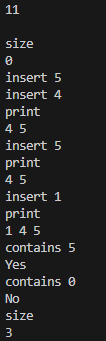


Рисунок 14: Приклад виконання програми №1



Рисунок 15: Факт прийняття розв’язку задачі середовищем Algotester

Час затрачений на виконання завдання - 1 день.

# **Висновки:**

*На цьому курсі я ознайомився з базовими принципами алгоритмізації та написання (консольних) програм на С++.*

**Епік 1**: Вивчив основи розробки та різноманітні середовища, це дозволило мені отримати уявлення про процес програмування та важливість правильного вибору середовища для ефективної розробки програмного забезпечення.

**Епік 2**: Ознайомився з основами програмування, включаючи лінійні та розгалужені алгоритми, оператори та типи даних, отримав тверді знання для написання ефективного коду та взаємодії з різними вбудованими функціями.

**Епік 3**: Оволодів роботою з циклами та функціями, включаючи рекурсію та вбудовані функції, зможу створювати більш структуровані програмні рішення, ефективніше перевикористовуватиму код.

**Епік 4**: Вивчав підходи, необхідні для роботи з масивами та структурами даних, включаючи вказівники та динамічні масиви, це дозволить мені створювати складні програми, які ефективно обробляють та зберігають дані.

**Епік 5**: Опанував навички роботи з файлами, включаючи роботу з текстовими та бінарними файлами. Вивчив деякі стандартні С- та С++-подібні методи для ефективнішої роботи зі стрічками.

**Епік 6**: Вивчив базові поняття в області динамічних структур даних, оскільки використання алгоритмів, які їх застосовують, сприяє створенню ефективних програм, які динамічно впорядковують та обробляють дані в реальному часі.