Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**Звіт**

про виконання

**Лабораторних та практичних робіт 5**

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

***з розділу***: «Файли. Системи числення. Бінарні Файли. Символи і Рядкові Змінні та Текстові Файли. Стандартна бібліотека та деталі/методи роботи з файлами. Створення й використання бібліотек»

***Виконав:***

студент групи ШІ-13

Золотарчук Максим

# **Тема роботи:**

Робота з файлами у мові програмування C++

# **Мета роботи:**

Ознайомитися з основними поняттями та операціями, пов’язаними з файлами у мові програмування C++, та навчитися створювати та використовувати власні бібліотеки для роботи з файлами

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

- Тема №1: Файли

- Тема №2: Системи числення

- Тема №3: Бінарні Файли

- Тема №4: Створення й використання бібліотек

1. Індивідуальний план опрацювання теорії:

* Тема №1: Файли.
  + Джерела Інформації
    - Стаття 1 - [C++ Files](https://www.w3schools.com/cpp/cpp_files.asp).
    - Стаття 2 - [How to append text to a text file in c++](https://stackoverflow.com/questions/2393345/how-to-append-text-to-a-text-file-in-c)
    - Стаття 3 - <https://icpc.upc.edu.cn/CLibrary/iostream/streampos.html>
    - Курс: Beginning C++ Programming - From Beginner to Beyond
  + Що опрацьовано:
    - Як працювати з файлами у C++ за допомогою класів std::fstream, std::ifstream, std::ofstream. Методи для відкриття, закриття, читання, запису, переміщення курсору та контролю стану файлів.
    - Різні формати та режими файлів у C++, такі як текстові, бінарні, додавання, перезапис, читання тощо. Параметри для вказання типу та способу доступу до файлів при відкритті.
    - Як обрати найкращий формат та режим файлу для своєї задачі. Як опрацьовувати різні типи даних у файлах, такі як символи, рядки, числа, структури тощо.
    - Як перевіряти, чи файл відкритий, чи немає помилок, чи досягнуто кінця файлу тощо. Функції для перевірки стану файлу, такі як is\_open(), good(), eof(), fail().
    - Бібліотека fstream, яка надає класи та функції для роботи з файлами у C++. Огляд основних можливостей та прикладів використання бібліотеки fstream.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 04.12.2023
  + Звершення опрацювання теми: 12.12.2023
* Тема №2: Системи числення.
  + Джерела Інформації:
    - Відео 1 -[Системи численн | Двійкова арифметика](https://www.youtube.com/watch?v=ai65R0WGq2Q&ab_channel=homeofourelectronics)
    - Стаття 1 - [Системи числення](https://ru.stackoverflow.com/questions/1466656/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0-%D1%81%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F-%D0%BD%D0%B0-%D1%81)
    - Курс: Beginning C++ Programming - From Beginner to Beyond
  + Що опрацьовано:
    - Визначення та приклади різних систем числення, таких як десяткова, двійкова, вісімкова та шістнадцяткова.
    - Правила та методи перетворення чисел між різними системами числення, такі як ділення з остачею, множення на основу, групування бітів або використання стандартних функцій бібліотеки c++.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 04.12.2023
  + Звершення опрацювання теми: 13.12.2023
* Тема №3: Бінарні Файли
  + Джерела Інформації:
    - Стаття 1 - [Binary file: write/read (c++)](https://ru.stackoverflow.com/questions/420525/%D0%91%D0%B8%D0%BD%D0%B0%D1%80%D0%BD%D1%8B%D0%B9-%D1%84%D0%B0%D0%B9%D0%BB-%D0%B7%D0%B0%D0%BF%D0%B8%D1%81%D1%8C-%D1%87%D1%82%D0%B5%D0%BD%D0%B8%D0%B5-%D1%81)
    - Стаття 2 - [Робота з бінарним файлом - C/C++](https://replace.org.ua/topic/7901/)
    - Курс - Beginning C++ Programming - From Beginner to Beyond
  + Що опрацьовано:
    - Відкриття та Закриття Файлу. Читання та Запис Даних. Перевірка Успішності Операцій. Позиціонування в Файлі.
    - Формати бінарних файлів: які переваги та недоліки бінарних файлів порівняно з текстовими, які стандарти та конвенції існують для бінарних файлів, як читати та записувати різні типи даних у бінарних файлах, як враховувати розмір та порядок байтів.
    - Серіалізація та десеріалізація об’єктів у C++: як зберігати та відновлювати стан об’єктів у бінарних файлах, як використовувати fstream для цього, які проблеми та обмеження можуть виникнути при серіалізації.
    - Перетворення типів покажчиків при читанні та записуванні даних, оскільки read та write працюють з char\* покажчиками:
  + Статус: Ознайомлений
  + Початок опрацювання теми: 05.12.2023
  + Звершення опрацювання теми: 09.12.2023
* Тема №4 Створення й використання бібліотек.
  + Джерела Інформації:
    - Відео 1 - [Write Better Code! | How to Create Shared Libraries in C/C++](https://www.youtube.com/watch?v=Slfwk28vhws&ab_channel=LowLevelLearning).
    - Стаття 1 - [How do I create a library?](https://stackoverflow.com/questions/16693273/how-do-i-create-a-library).
    - Курс - Beginning C++ Programming - From Beginner to Beyond
  + Що опрацьовано:
    - Статичні, динамічні та заголовкові бібліотеки
    - Розглянуто можливість створення влсних бібліотек за допомогою компілятора, IDE, або інших інструментів
  + Статус: Ознайомлений
  + Початок опрацювання теми: 04.12.2023
  + Звершення опрацювання теми: 07.12.2023

# **Виконання роботи:**

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

Завдання №1 VNS Lab №6

* Варіант завдання – 3
* Надрукувати всі слова, які містять по одній цифрі.Деталі завдання

Завдання №2 VNS Lab №8

* Варіант завдання – 3

Структура "Держава":

- назва;

- столиця;

- чисельність населення;

- займана площа.

Знищити всі елементи, у яких чисельність менше заданої, додати елемент

після елемента із зазначеним номером..

* Деталі завдання

Сформувати двійковий файл із елементів, заданої у варіанті структури,

роздрукувати його вмістиме, виконати знищення й додавання елементів у

відповідності зі своїм варіантом, використовуючи для пошуку елементів що

знищуються чи додаються, функцію. Формування, друк, додавання й знищення

елементів оформити у вигляді функцій.

* Важливі деталі для врахування в імплементації програми

Використання вище наведеної теорії для імплементації програми.

Передбачити повідомлення про помилки при відкритті файлу й виконанні операцій вводу/виводу.

Завдання №3 VNS Lab №9

* Варіант завдання - 3
* Деталі завдання

Створити текстовий файл F1 не менше, ніж з 10 рядків і записати в нього

1) Скопіювати у файл F2 тільки ті рядки з F1, які починаються й закінчуються

на ту саму букву.

2) Підрахувати кількість символів в F2.

Завдання №4 Class Practice Files 1

* Деталі завдання

Запис текстової стрічки у файл із заданим ім’ям

Реалізувати функцію створення файла і запису в нього даних:

enum FileOpResult { Success, Failure, … };

FileOpResult write\_to\_file(char \*name, char \*content);

Умови задачі:

- створити файл із заданим ім’ям; якщо файл існує – перезаписати його вміст

- написати код стійкий до різних варіантів вхідних параметрів

- name – ім’я, може не включати шлях

- записати у файл вміст стрічки content, прочитати content із стандартного вводу

- повернути статус операції: Success – все пройшло успішно, Failure – файл не вдалося створити, або збій операції відкриття, запису даних, чи закриття файла.

* Важливі деталі для врахування в імплементації програми

Завдання №5 Class Practice Files 2

* Деталі завдання

Копіювання вмісту файла у інший файл

Реалізувати функцію створення файла і запису в нього даних:

enum FileOpResult { Success, Failure, … };

FileOpResult copy\_file(char \*file\_from, char \*file\_to);

Умови задачі:

- копіювати вміст файла з ім’ям file\_from у файл з ім’ям file\_to; написати код стійкий до різних варіантів вхідних параметрів, обробити всі можливі варіанти відсутності одного з файлів

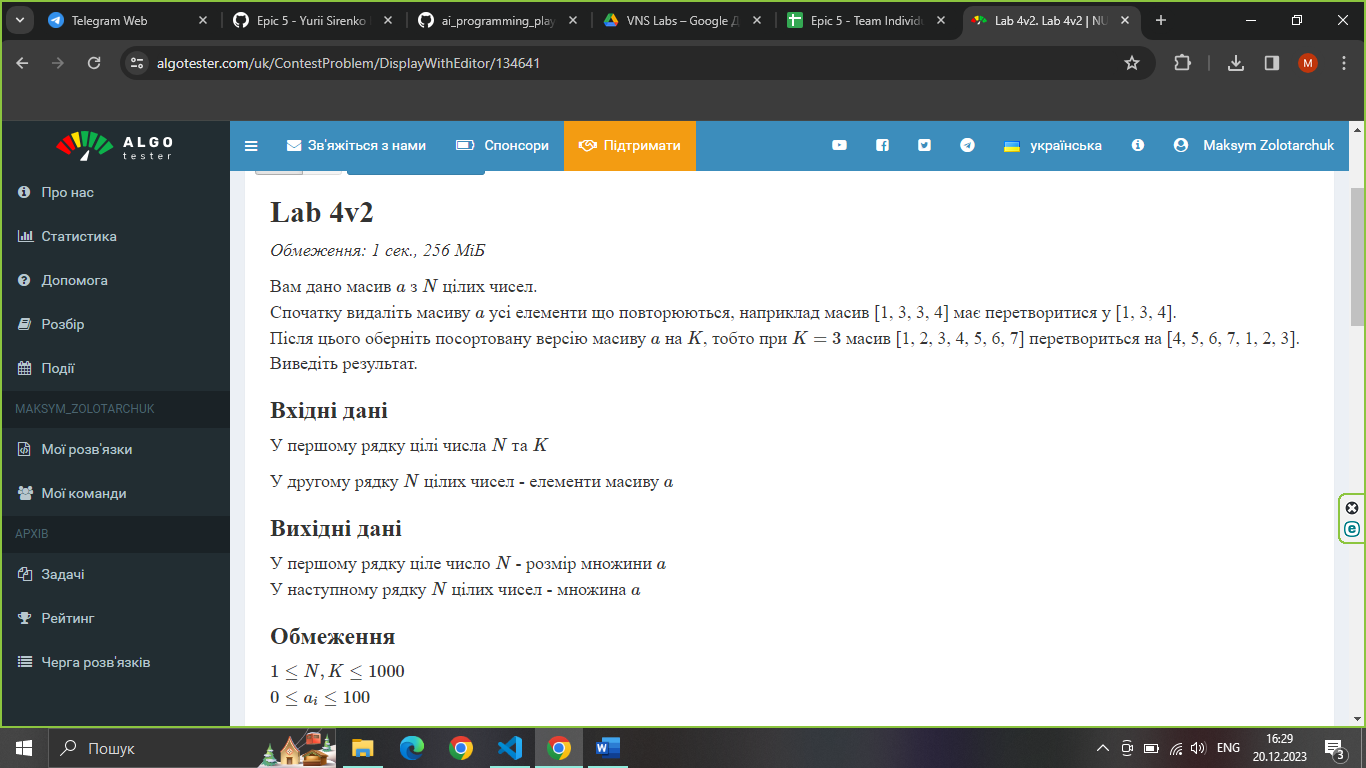
- file\_from, file\_to – можуть бути повним або відносним шляхом

- повернути статус операції: Success – все пройшло успішно, Failure – файл не вдалося створити, або збій операції відкриття, читання чи запису даних, закриття файла.

* Важливі деталі для врахування в імплементації програми

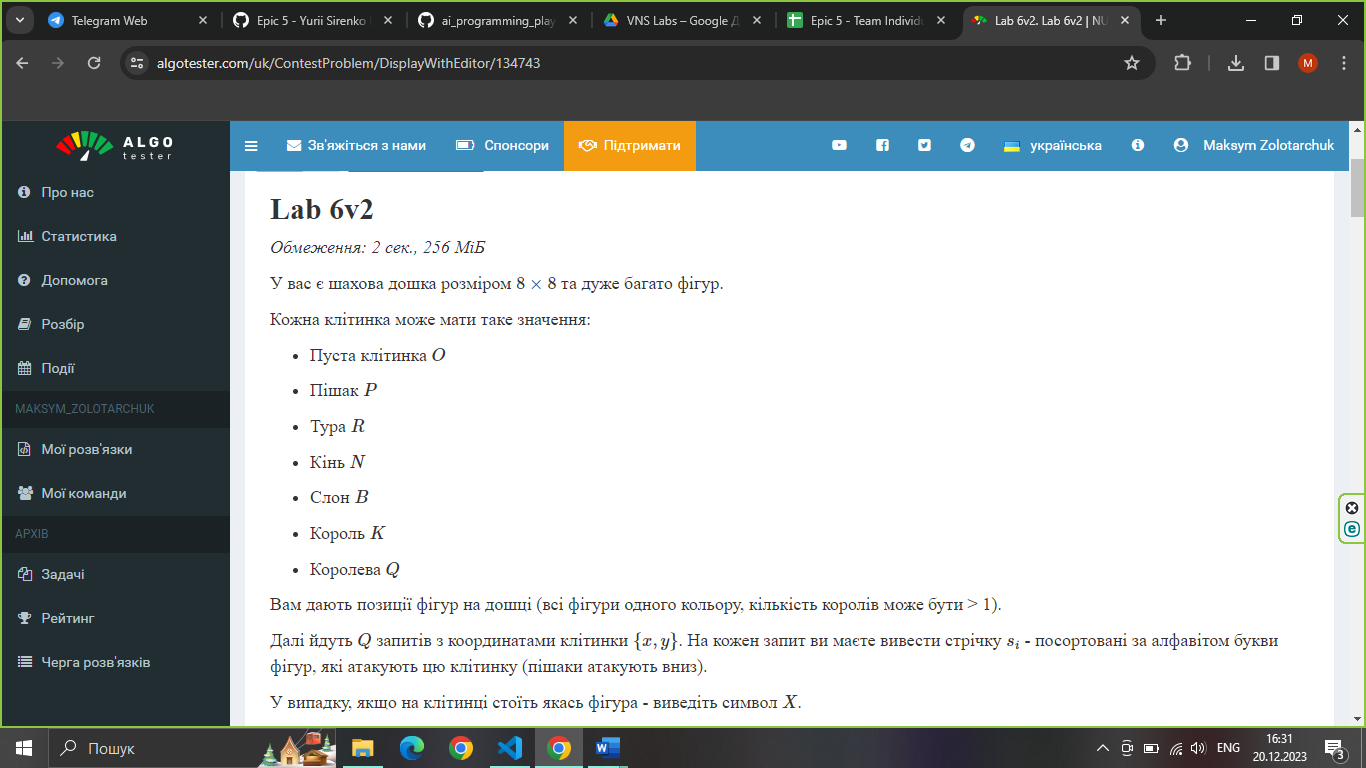
Завдання №6 Algotester Lab №4

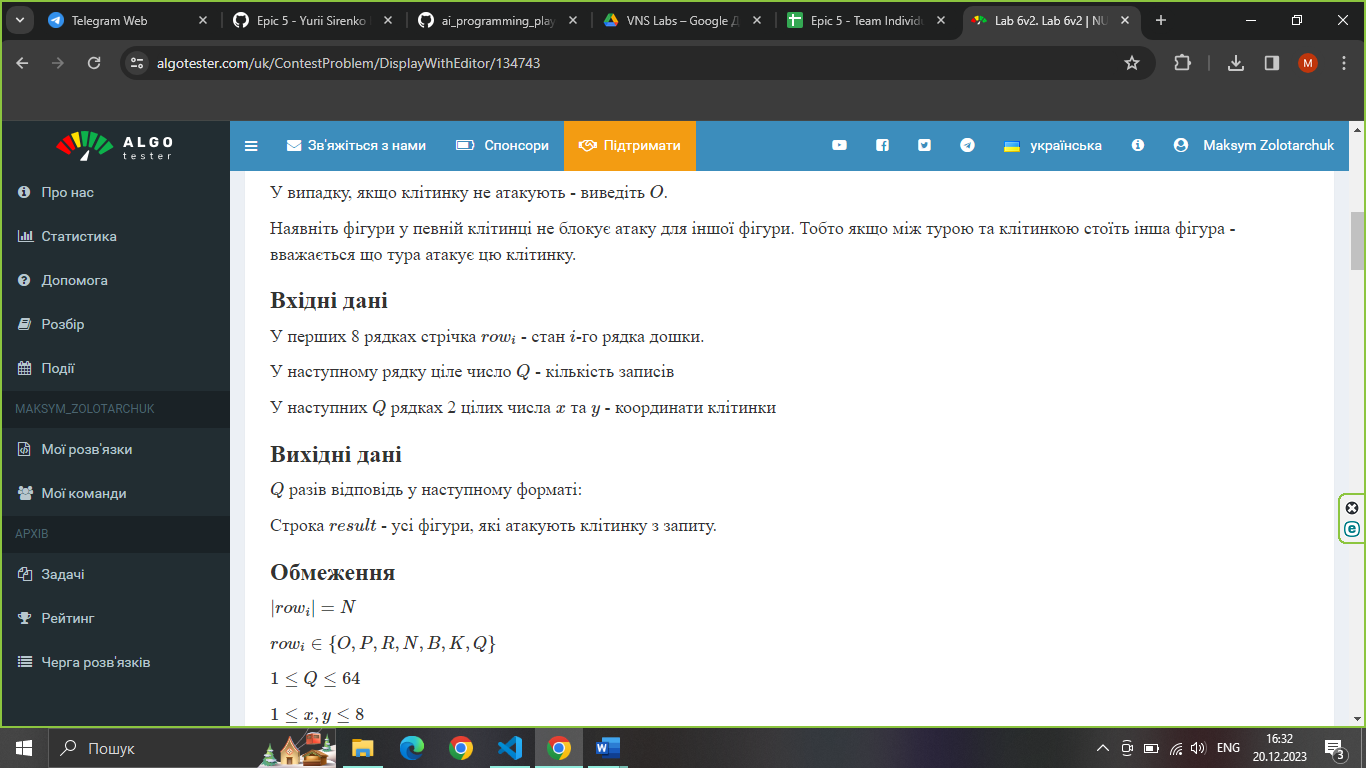
* Варіант завдання – 2



Завдання №7 Algotester Lab №6

* Варіант завдання – 3





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

Програма №1 - VNS Lab №6

* Планований час на реалізацію – 2 година
* Час затрачений на виконання завдання – 3 години

Програма №2 - VNS Lab №8

* Планований час на реалізацію – 4 година
* Час затрачений на виконання завдання – 1 доба

Програма №3 - VNS Lab №9

* Планований час на реалізацію – 6 година
* Час затрачений на виконання завдання – 8 година

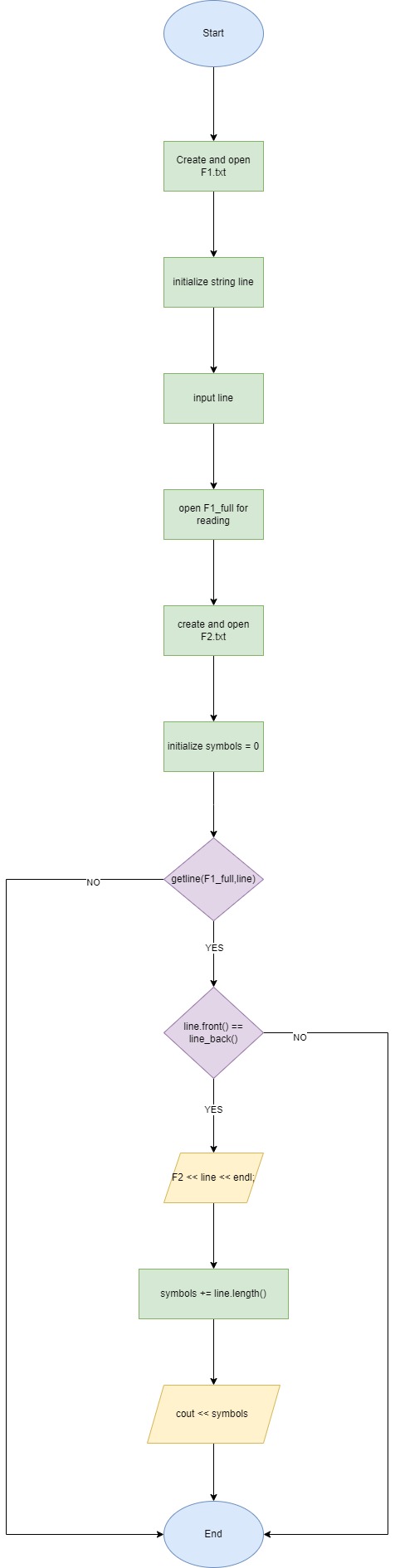


Рисунок Блок-схема до VNS Lab №9

Програма №4 - Class Practice Files 1

* Планований час на реалізацію – 1 година
* Час затрачений на виконання завдання – 40 хвилин

Програма №5 - Class Practice Files 2

* Планований час на реалізацію – 1 година
* Час затрачений на виконання завдання – 20 хвилин

Програма №6 - Algotester Lab №4

* Планований час на реалізацію – 3 години
* Час затрачений на виконання завдання – 3 години

Програма №7 - Algotester Lab №6

* Планований час на реалізацію – 3 години
* Час затрачений на виконання завдання – 5 годин

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

Завдання №1 - VNS Lab №6

#include <iostream>

#include <string>

#include <sstream> // istringstream

using namespace std;

int main()

{

*string* token;

    cout << "Enter a string: " << endl;

    getline(cin, token);

*istringstream* iss(token);

*string* word;

    int numbers = 0;

    while (iss >> word)

    {

        for (int c : word)

        {

            if (isdigit(c))

            {

                ++numbers;

            }

        }

    }

    if (numbers == 1)

    {

        cout << word;

    }

    else

    {

        cout << "This word doesn't have number" << endl;

    }

    return 0;

}

*Код програми №1 - VNS Lab №6*

Завдання №2 - VNS Lab №8

#include <iostream>

#include <string>

#include <vector>

#include <fstream>

using namespace std;

struct *State*

{

*string* name;

*string* capital;

 long long int population;

 long long int occupied\_area;

};

void printState(const *State*& *state*)

{

 cout << "Name: " << *state*.name << endl;

 cout << "Capital: " << *state*.capital << endl;

 cout << "Population: " << *state*.population << endl;

 cout << "Occupied Area: " << *state*.occupied\_area << endl;

}

void writetobinfile(const *string*& *filename*, const vector<*State*>& *states*)

{

*ofstream* file(*filename*, *ios*::binary);

 if (!file)

 {

  cerr << "Cannot open file!" << endl;

  return;

 }

 for (const *State*& state : *states*)

 {

  file.write(reinterpret\_cast<const char\*>(&state), sizeof(*State*)); //використовую write для запису в бінарний файл

 }

}

void deletePopulation(vector<*State*>& *states*, long long int *set\_population*)

{

 for (int i = 0; i < *states*.size();)

 {

  if (*states*[i].population < *set\_population*)

  {

*states*.erase(*states*.begin() + i);

  }

  else

  {

   ++i;

  }

 }

}

 void addState(vector<*State*>&*states*, int *id*, const *State* & *newState*)

 {

  if (*id* >= 0 && *id* < *states*.size())

  {

*states*.insert(*states*.begin() + *id* + 1, *newState*);

  }

  else

  {

*states*.push\_back(*newState*);

  }

 }

int main()

{

 std::vector<*State*> states =

 {

    {"Ukraine", "Kyiv", 43000000, 603628},

    {"Poland", "Warsaw", 37750000, 322575},

    {"Israel", "Jerusalem", 9364000, 22145},

    {"China", "Beijing", 1412000000, 9597000}

 };

 writetobinfile("States.bin", states);

 cout << "First edition: " << endl;

 for (const auto& state : states)

 {

  printState(state);

 }

 cout << endl;

 deletePopulation(states, 10000000);

*State* newState = { "Turkey", "Ankara", 84780000, 783562 };

 addState(states, 1, newState);

 writetobinfile("States.bin", states);

 cout << "Second edition: " << endl;

 for (const auto& state : states)

 {

  printState(state);

 }

 return 0;

}

*Код програми №2 - VNS Lab №8*

Завдання №3 - VNS Lab №9

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

int main()

{

*ofstream* F1("D:/algorithms/F1.txt");

*string* line;

  cout << "Enter in first file (F1): " << endl;

  for (int i = 0; i < 10; ++i)

  {

   getline(cin, line);

   F1 << line << endl;

  }

*ifstream* F1\_full("D:/algorithms/F1.txt");

*ofstream* F2("D:/algorithms/F2.txt");

  int symbols = 0;

  while (getline(F1\_full, line))

  {

   if (line.front() == line.back())

   {

    F2 << line << endl;

    symbols += line.length();

   }

  }

  cout << "Amount of symbls in F2: " << symbols;

  return 0;

}

*Код програми №3 - VNS Lab №9*

Завдання №4 - Class Practice 1

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

// Оголошення перелічення для результату операції з файлом

enum class *FileOpResult*

{

    Success,

    Failure

};

// Функція для запису вмісту в файл

*FileOpResult* writeToFile(const *string* &*name*, const *string* &*content*)

{

    // Використання конструктора для відразу відкриття файлу

*ofstream* file(*name*);

    // Перевірка, чи вдалося відкрити файл

    if (!file.is\_open())

    {

        return *FileOpResult*::Failure; // Повернення Failure при невдалому відкритті

    }

    // Запис вмісту в файл

    file << *content*;

    // Перевірка, чи вдалося записати в файл

    if (!file)

    {

        return *FileOpResult*::Failure; // Повернення Failure при помилці запису

    }

    // Файл автоматично закриється при знищенні об'єкта file

    return *FileOpResult*::Success; // Повернення Success при успішному відкритті та записі

}

int main()

{

*string* filename = "test.txt";

*string* content = "Це приклад вмісту файлу.";

*FileOpResult* result = writeToFile(filename, content);

    if (result == *FileOpResult*::Success)

    {

        cout << "Success\n";

    }

    else

    {

        cout << "Error" << endl;;

    }

    return 0;

}

*Код програми №4 – Class Practice 1*

Завдання №5 - Class Practice 2

Посилання на файл програми у пул-запиті GitHub

#include <fstream>

#include <iostream>

#include <string>

using namespace std;

enum class *OperationResult*

{

    Success,

    Failure

};

*OperationResult* copyFileContents(const *string* &*sourcePath*, const *string* &*destinationPath*)

{

    if (*sourcePath*.empty() || *destinationPath*.empty())

    {

        cout << "Error: Source or destination path is empty." << endl;

        return *OperationResult*::Failure;

    }

*ifstream* sourceFile(*sourcePath*, *ios*::binary);

    if (!sourceFile)

    {

        cout << "Error: Unable to open source file at " << *sourcePath* << endl;

        return *OperationResult*::Failure;

    }

*ofstream* destinationFile(*destinationPath*, *ios*::binary);

    if (!destinationFile)

    {

        cout << "Error: Unable to open or create destination file at " << *destinationPath* << endl;

        return *OperationResult*::Failure;

    }

    destinationFile << sourceFile.rdbuf();

    if (!destinationFile)

    {

        cout << "Error: Failure during the file copy process." << endl;

        return *OperationResult*::Failure;

    }

    return *OperationResult*::Success;

}

int main()

{

    const *string* sourceFilePath = "source.txt";

    const *string* destinationFilePath = "destination.txt";

*OperationResult* result = copyFileContents(sourceFilePath, destinationFilePath);

    if (result == *OperationResult*::Success)

    {

        cout << "File was successfully copied.\n";

    }

    else

    {

        cout << "Error occurred during file copy.\n";

    }

    return 0;

}

*Код програми №5 – Class Practice 2*

Завдання №6 - Algotester Lab №4

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

int main()

{

 int N = 0;

 int K = 0;

 cin >> N >> K;

 vector<int> a(N);

 for (int i = 0; i < N; ++i)

 {

  cin >> a[i];

 }

 sort(a.begin(), a.end());

 a.erase(unique(a.begin(), a.end()), a.end());

 K %= a.size();

 rotate(a.begin(), a.begin() + K, a.end());

 cout << a.size() << endl;

 for (int i : a)

 {

  cout << i << " ";

 }

 return 0;

}

*Код програми №6 – Algotester Lab №4*

Завдання №7 - Algotester Lab №6

#include <iostream>

#include <vector>

#include <string>

#include <algorithm>

using namespace std;

struct *Figure*

{

    char type;

    int row;

    int col;

    Figure(char *t*, int *r*, int *c*) : type(*t*), row(*r*), col(*c*) {}

};

char isUnderAttack(const *Figure*& *piece*, int *x*, int *y*)

{

    int a = *piece*.row + 1;

    int b = *piece*.col + 1;

    if (*x* == a && *y* == b)

    {

        return 'X';

    }

    switch (*piece*.type)

    {

    case 'P':

        return (*x* == a + 1 && (*y* == b + 1 || *y* == b - 1)) ? 'P' : 0;

    case 'R':

        return (*x* == a || *y* == b) ? 'R' : 0;

    case 'B':

        return (abs(*x* - a) == abs(*y* - b)) ? 'B' : 0;

    case 'N':

        return ((abs(*x* - a) == 2 && abs(*y* - b) == 1) || (abs(*x* - a) == 1 && abs(*y* - b) == 2)) ? 'N' : 0;

    case 'Q':

        return (*x* == a || *y* == b  || abs(*x* - a) == abs(*y* - b)) ? 'Q' : 0;

    case 'K':

        return (abs(*x* - a) <= 1 && abs(*y* - b) <= 1) ? 'K' : 0;

    default:

        return 0;

    }

}

bool isInvalid(char *c*)

{

    return *c* == 'R' || *c* == 'P' || *c* == 'R' || *c* == 'B' || *c* == 'N'||  *c* == 'Q' || *c* == 'K' || *c* == 'X';

}

bool check(const *string*& *str*)

{

    return all\_of(*str*.begin(), *str*.end(), [](char *c*) { return !isInvalid(*c*); });

}

void processQueries(const vector<*Figure*>& *pieces*, const vector<pair<int, int>>& *queries*) {

    for (const auto& query : *queries*)

    {

        int x = query.first;

        int y = query.second;

*string* answer;

        for (const *Figure*& piece : *pieces*)

        {

            char resultChar = isUnderAttack(piece, x, y);

            if (isalpha(resultChar)) {

                answer += resultChar;

            }

        }

        if (check(answer))

        {

            cout << 'O' << endl;

        }

        else {

*size\_t* position = answer.find('X');

            if (position != *string*::npos)

            {

                cout << "X" << endl;

            }

            else

{

                sort(answer.begin(), answer.end());

                auto it = unique(answer.begin(), answer.end());

                answer.erase(it, answer.end());

                cout << answer << endl;

            }

        }

    }

}

int main()

{

    vector<*Figure*> pieces;

    vector<pair<int, int>> queries;

    for (int i = 0; i < 8; i++)

    {

        for (int j = 0; j < 8; j++)

        {

            char type;

            cin >> type;

            if (type != 'O')

            {

                pieces.emplace\_back(type, i, j);

            }

        }

    }

    int q, x, y;

    cin >> q;

    for (int t = 0; t < q; t++)

    {

       cin >> x >> y;

        queries.emplace\_back(x, y);

    }

    processQueries(pieces, queries);

    return 0;

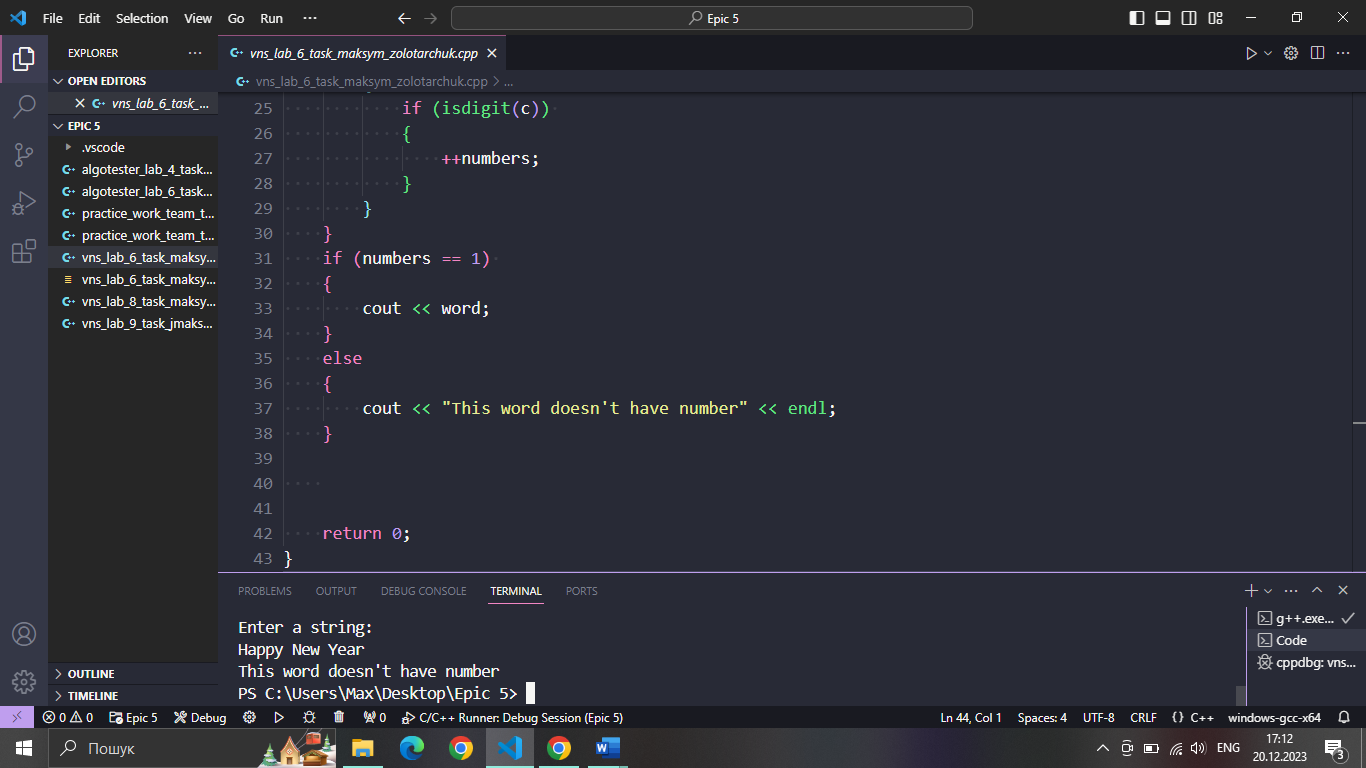
}

*Код програми №6 – Algotester Lab №6*

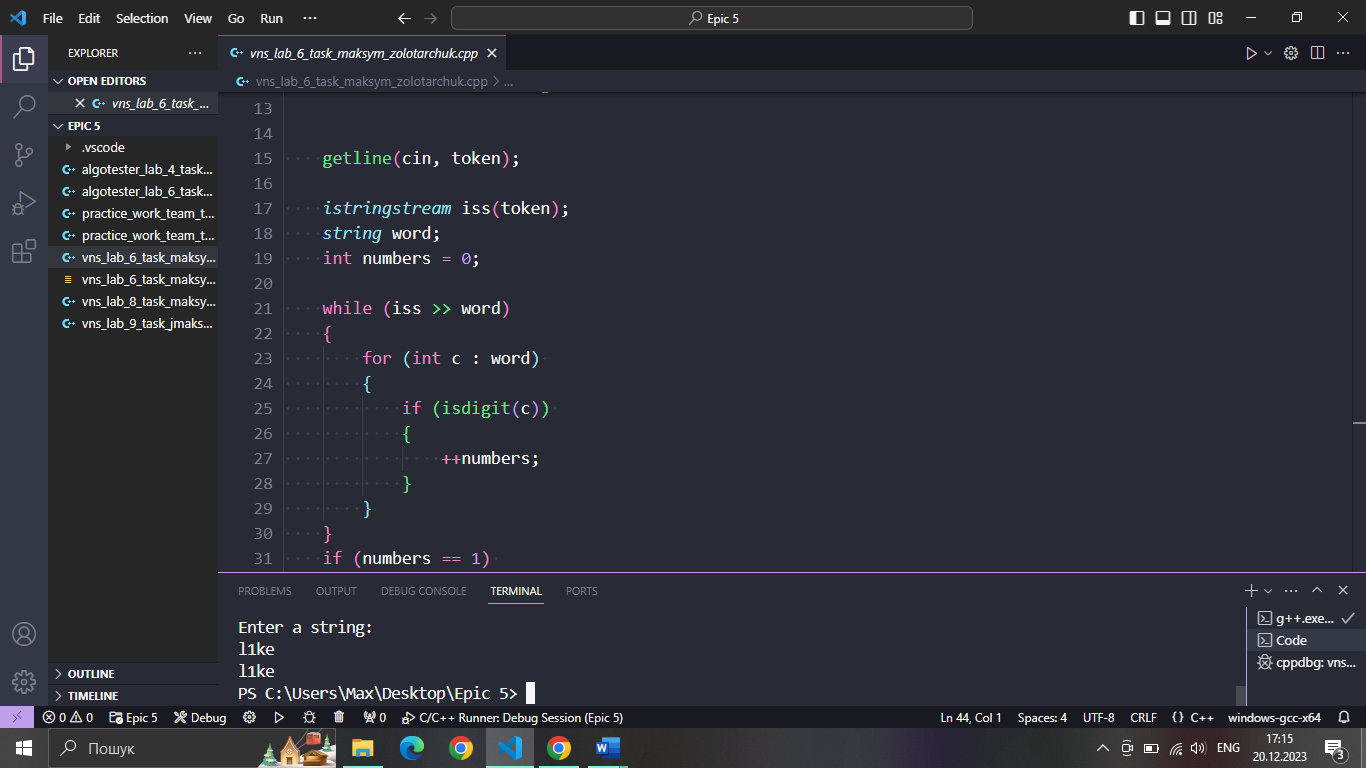
## **4. Результати виконання завдань, тестування та фактично затрачений час:**

Завдання №1 - VNS Lab №6

Деталі по виконанню і тестуванню програми



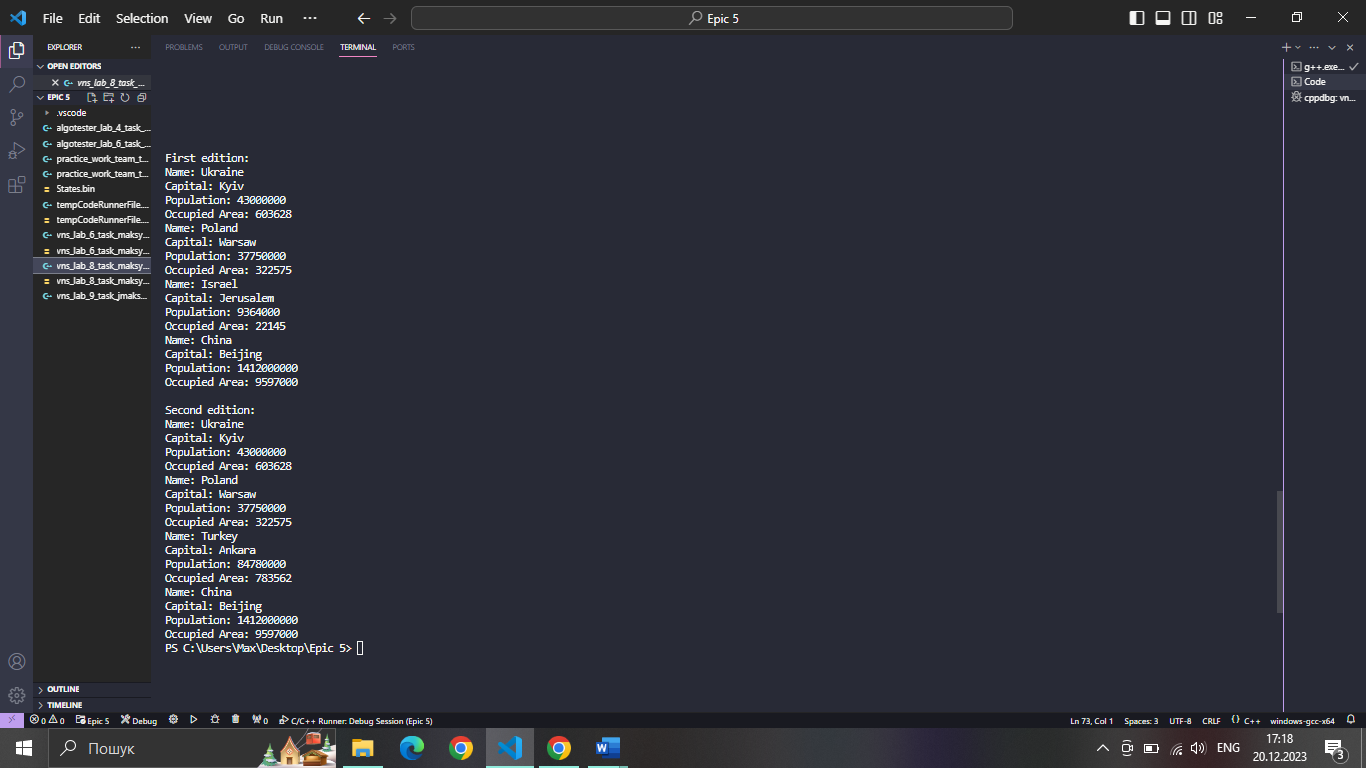
VNS Lab №6 test



VNS Lab №6 test

Завдання №2 - VNS Lab №8

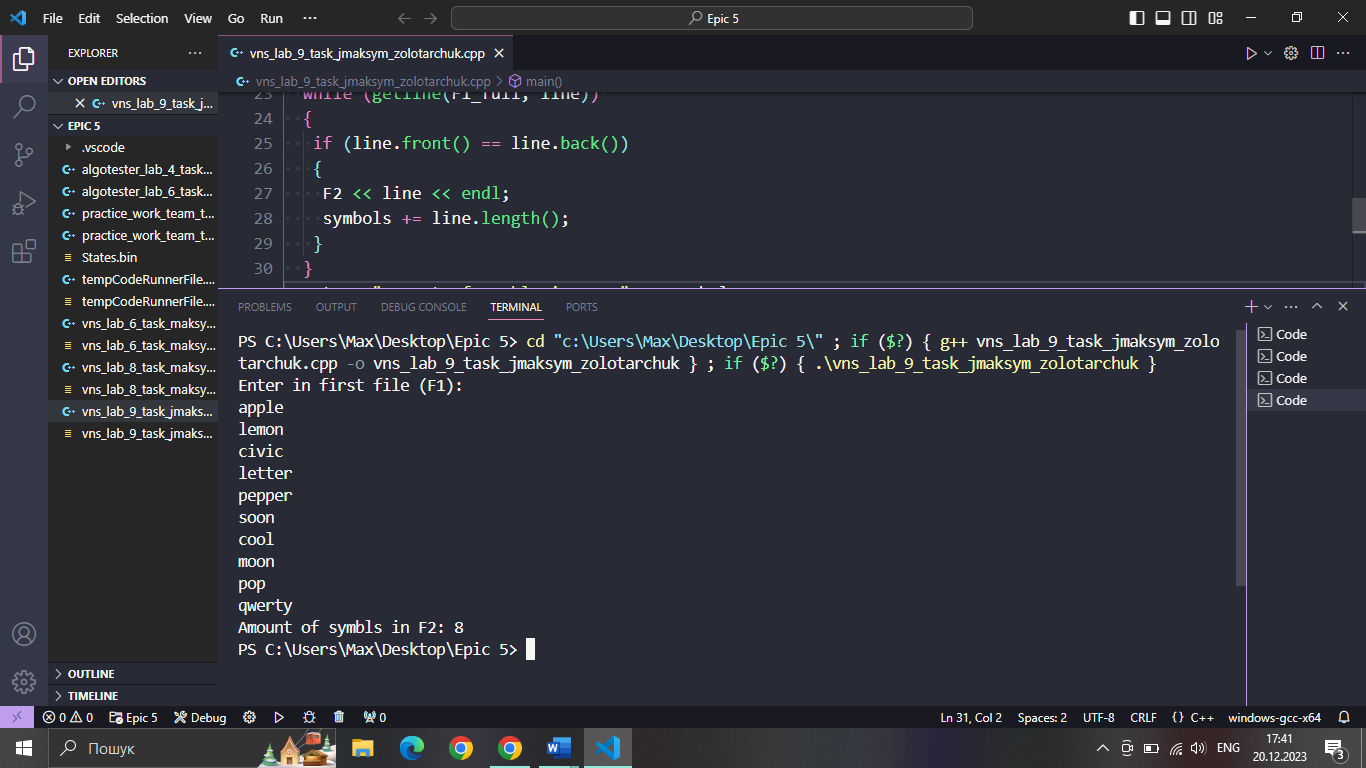
Деталі по виконанню і тестуванню програми



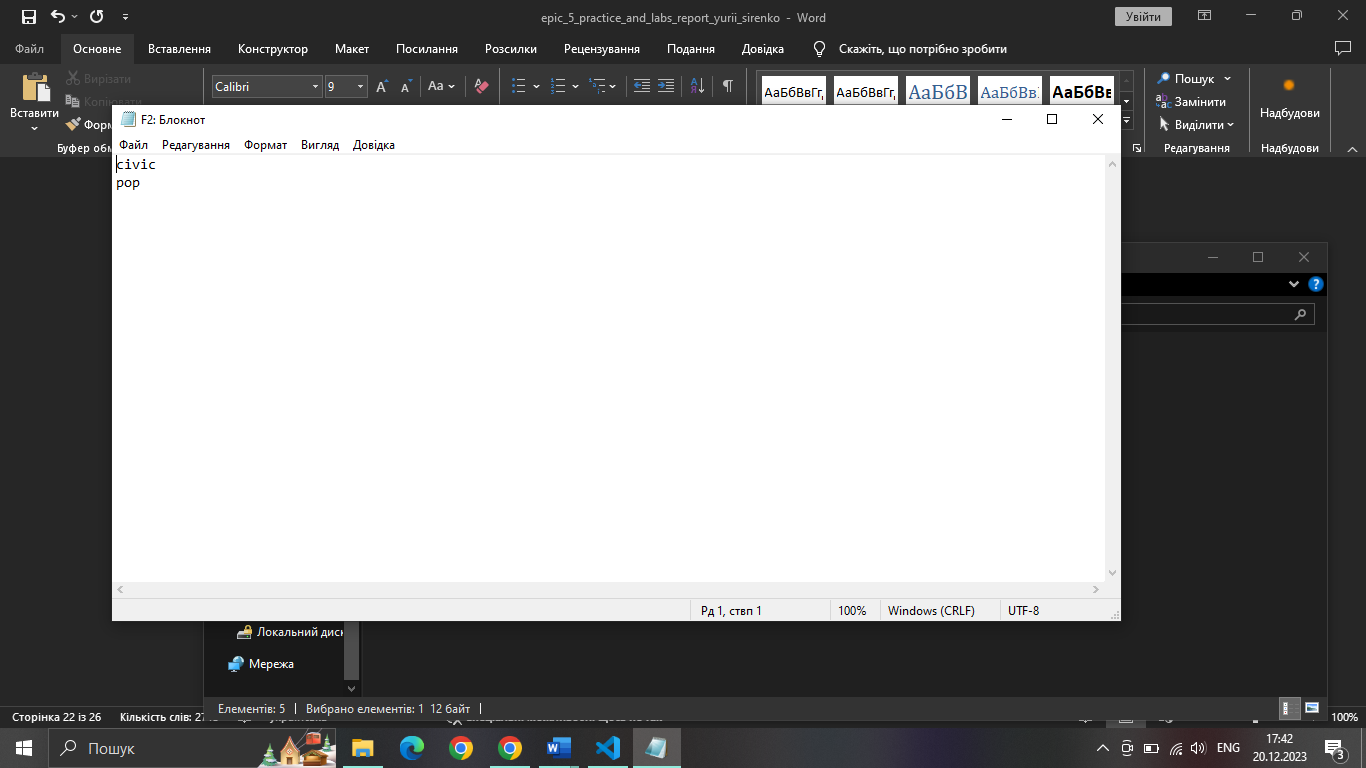
VNS Lab №8 test

Завдання №3 - VNS Lab №9

Деталі по виконанню і тестуванню програми

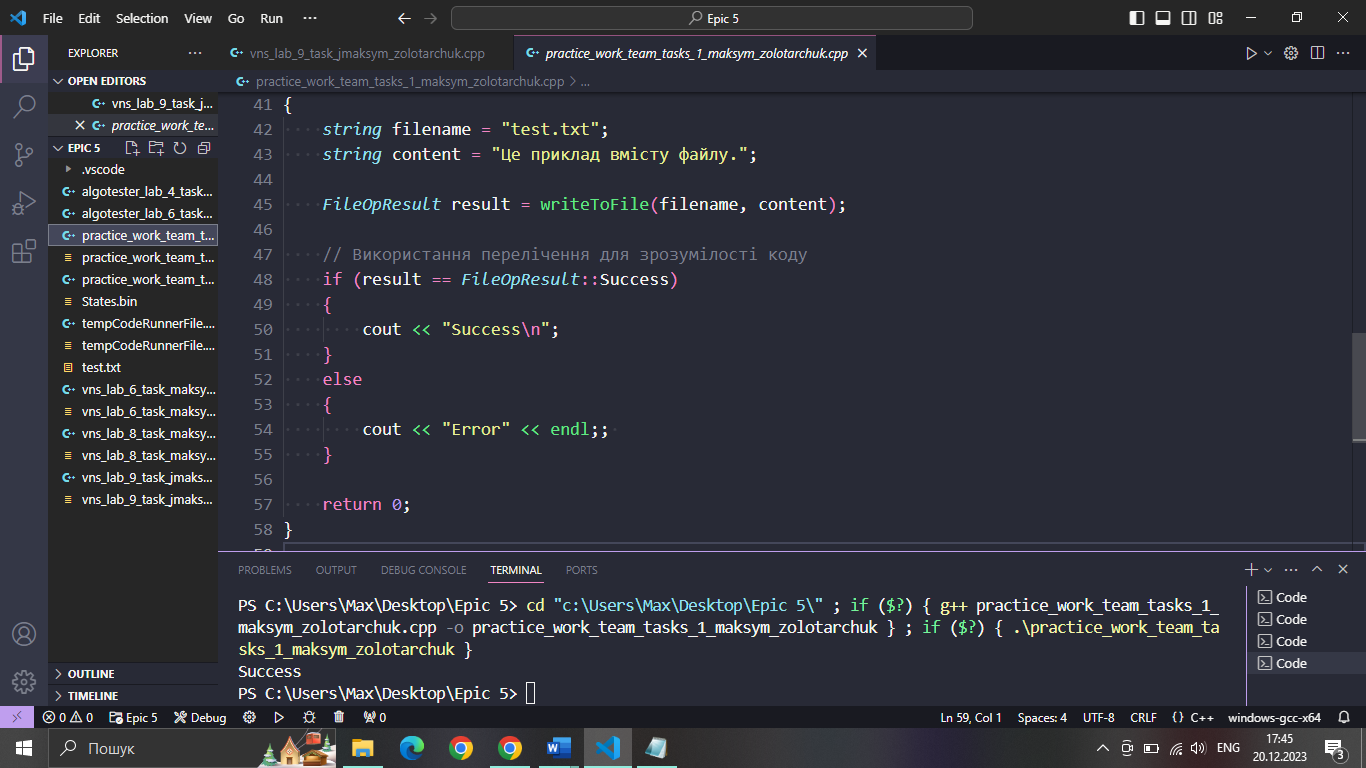


VNS Lab №9 test



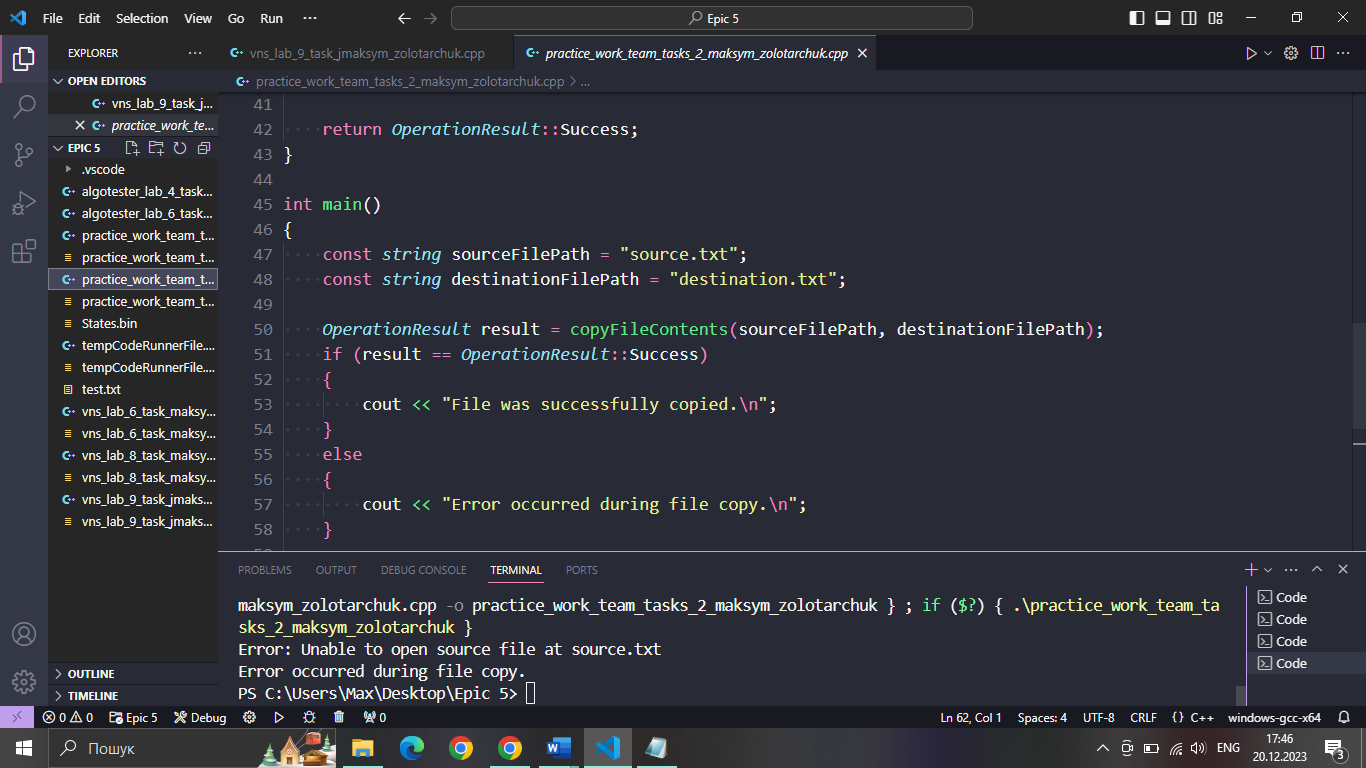
Завдання №4 - Class Practice Files 1

Деталі по виконанню і тестуванню програми



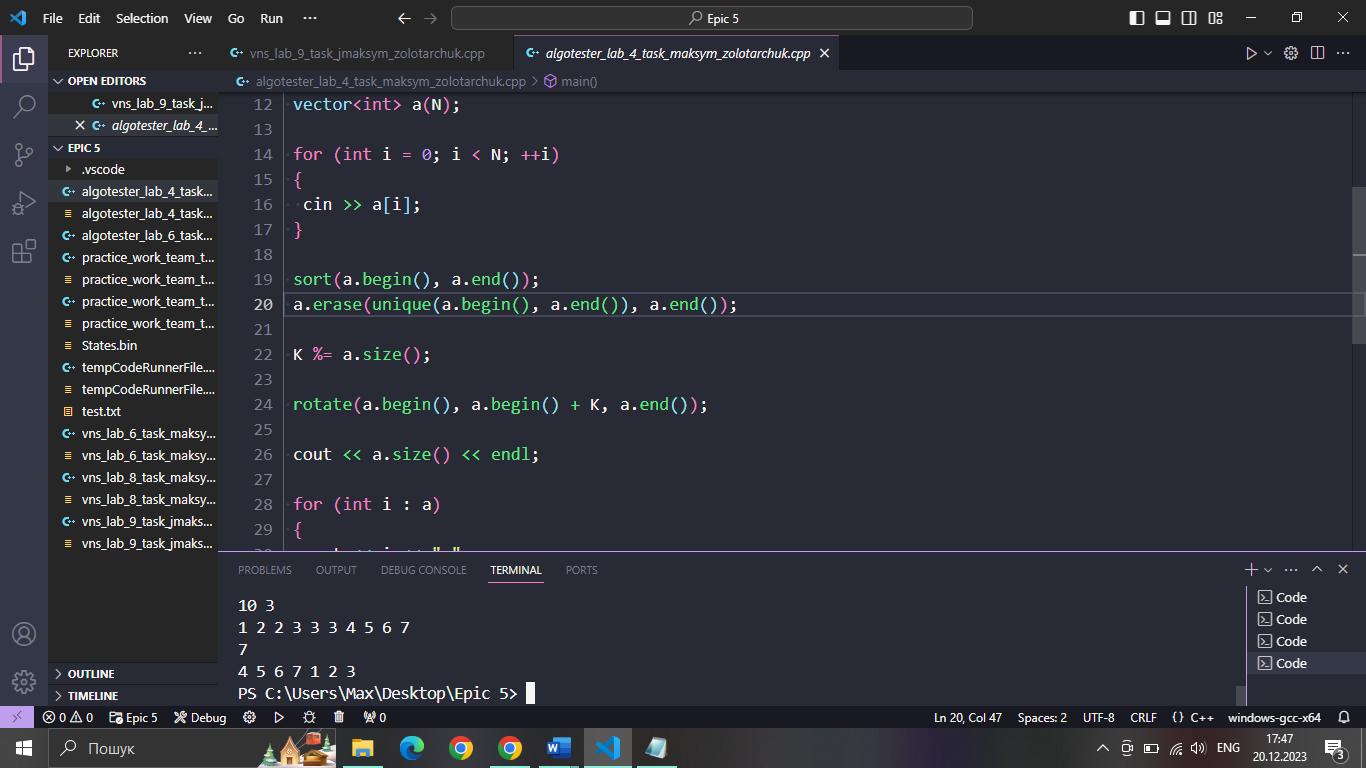
Class Practice Files

Завдання №5 - Class Practice Files 2

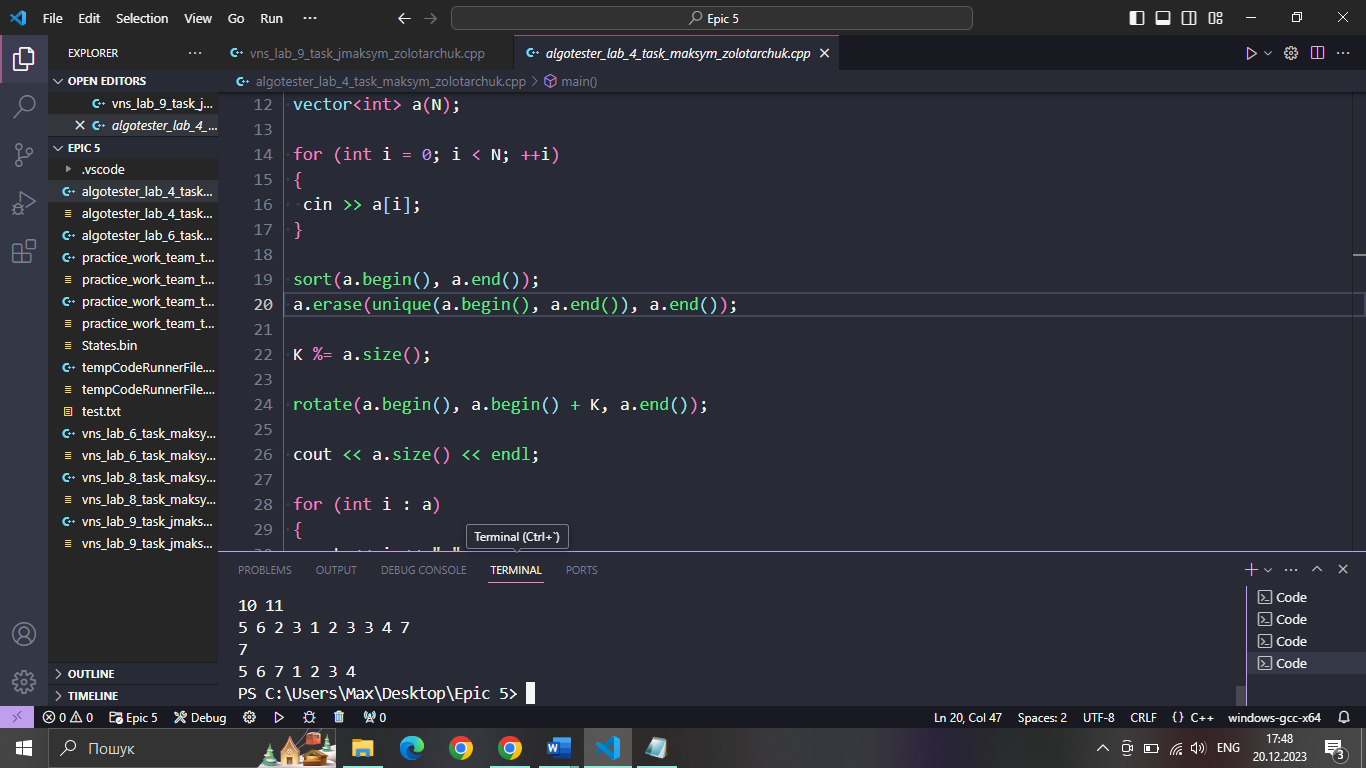


Class Practice Files 2

Завдання №6 - Algotester Lab №4

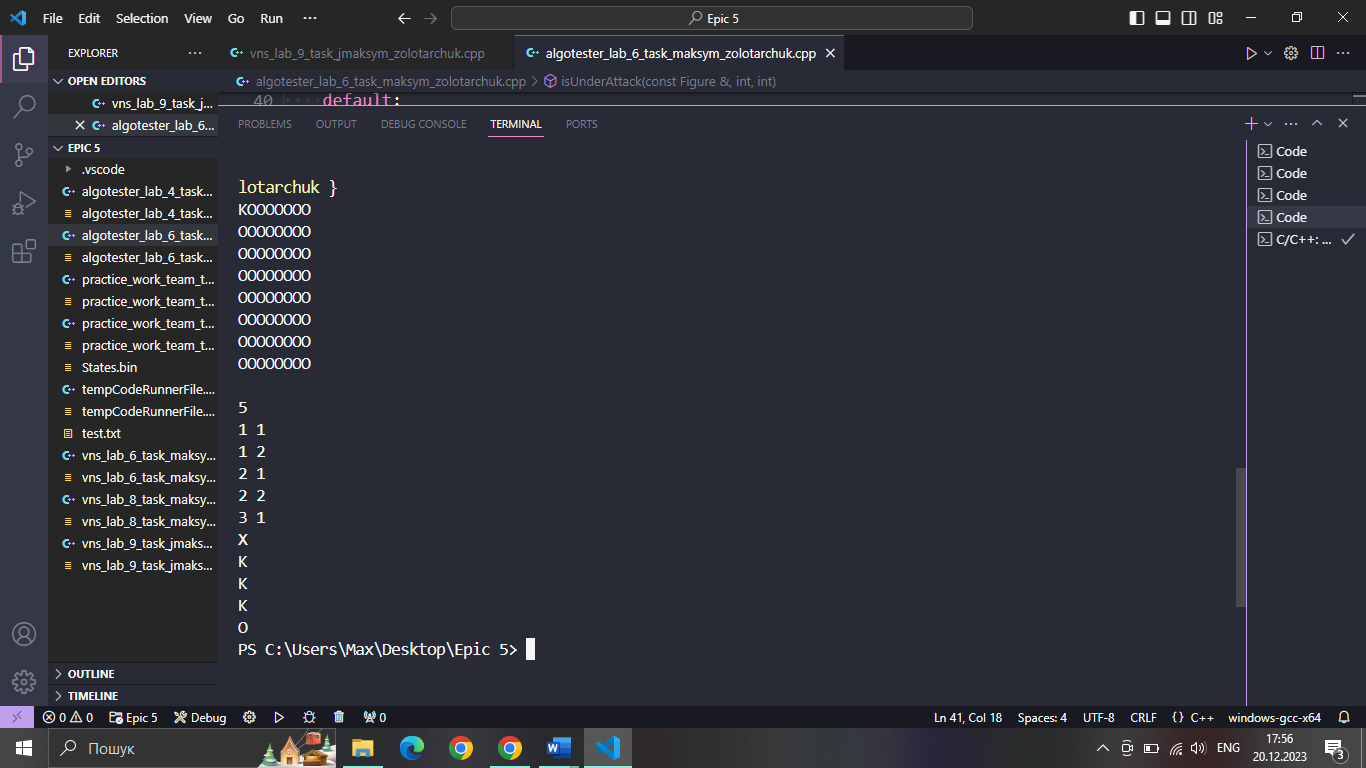
Деталі по виконанню і тестуванню програми 

Algotester Lab №4 test

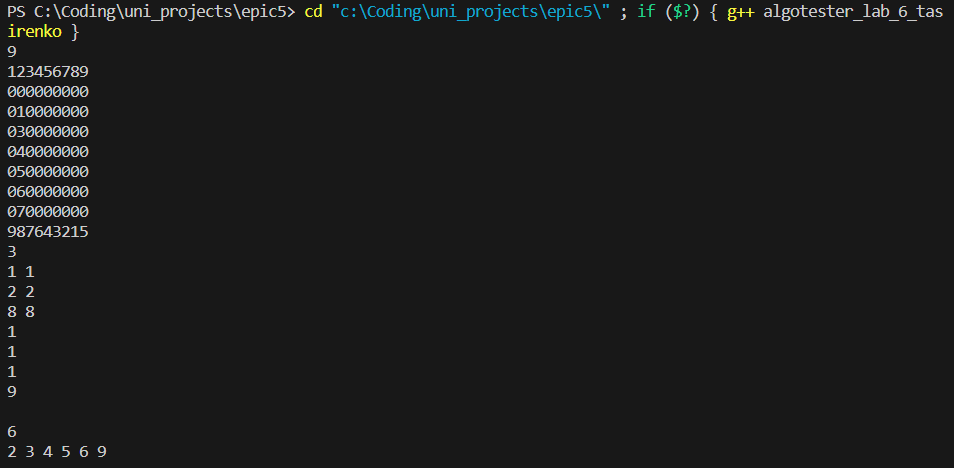


Завдання №7 - Algotester Lab №6

Деталі по виконанню і тестуванню програми

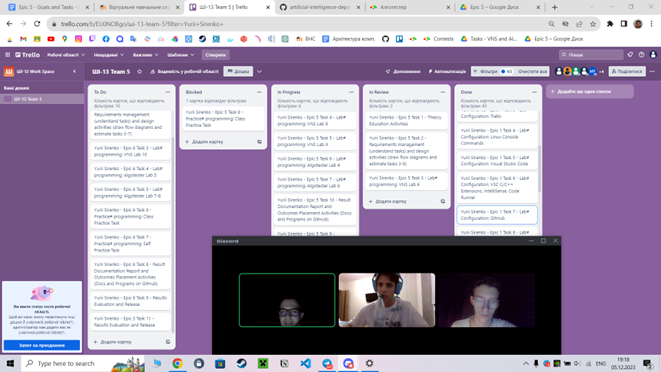
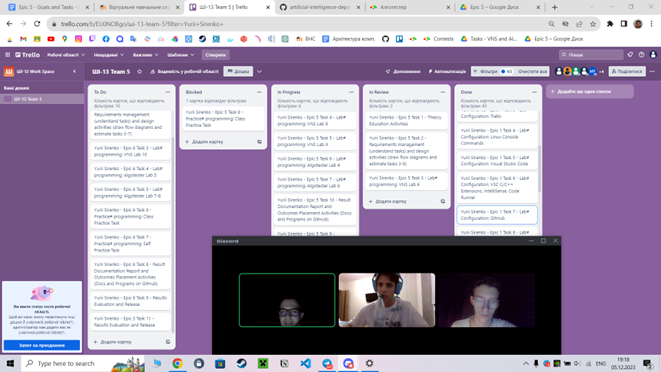


Algotester Lab №6 test



Algotester Lab №6 test 2

## **5. Кооперація з командою:**

* Скрін з 1-ї зустрічі по обговоренню задач Епіку та Скрін прогресу по Трелло
* 
* Скрін з 2-ї зустрічі по обговоренню задач Епіку та Скрін прогресу по Трелло
* 
* GitHub: <https://github.com/artificial-intelligence-department/ai_programming_playground/tree/epic_5_practice_and_labs_maksym_zolotarchuk/ai_13/maksym_zolotarchuk/Epic_5_Maksym_Zolotarchuk/Code>

# **Висновки:**

У процесі виконання лабораторної роботи з файлами та використання бібліотек у мові програмування C++, я отримав важливі навички та знання для ефективної роботи з файловою системою. Ознайомлення з основними концепціями, такими як робота з бінарними та текстовими файлами, системами числення, символами та рядками, дозволило мені глибше розібратися в механізмах обробки інформації в мові C++. Я особливу увагу приділив створенню та використанню власних бібліотек, що відкрило нові можливості для оптимізації та узагальнення коду при подальшому програмуванні. Ця лабораторна робота стала важливим етапом у моєму навчанні, дозволяючи мені поглибити розуміння принципів роботи з файлами та ефективного їх використання у програмах на C++.