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 № 5**

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

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

до:

ВНС Лабораторної Роботи № 6

ВНС Лабораторної Роботи № 8

ВНС Лабораторної Роботи № 9

Алготестер Лабораторної Роботи №4

Алготестер Лабораторної Роботи №6

Практичних Робіт №5

**Виконав:**

Студент групи ШІ-13

Булишин Віктор Михайлович

**Тема роботи:**

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

**Мета роботи:**

Task 1 - Theory Education Activities

Task 2 - Requirements management (understand tasks) and design activities (draw flow chart and estimate tasks 3-9)

Task 3 - programming: VNS Lab 6

Task 4 - programming: VNS Lab 8

Task 5 - programming: VNS Lab 9

Task 6 - programming: Algotester Lab 4

Task 7 - programming: Algotester Lab 6

Task 8 - programming: Class Practice Task

Task 10 - Result Documentation Report and Outcomes Placement Activities (Docs and Programs on GitHub)

Task 11 - Results Evaluation and Release

**Теоретичні відомості:**

* Теоретичні відомості з переліком важливих тем:
* Тема №1: Файли. Бінарні та текстові файли.
* Тема №2: Символи і рядкові змінні
* Тема №3: Стандартна бібліотека
* Індивідуальний план опрацювання теорії:
* Тема №1: Файли. Бінарні та текстові файли.
* Джерела Інформації.
* Відео.
* <https://youtu.be/rkJJzH-ymg8?si=Re0jBT47DOap8UXs>
* <https://youtu.be/8jLOx1hD3_o?si=56bJhdsRpmcvIQhx>

Стаття.

* <https://cplusplus.com/reference/fstream/fstream/>
* Що опрацьовано:
* Опрацьовано види файлів, їх особливості, поняття потоку та його види в С++.
* Статус: Ознайомлений
* Початок опрацювання теми: 14.12.2023
* Звершення опрацювання теми: 15.12.2023
* Тема №2: Символи і рядкові змінні
* Джерела Інформації:
* Відео.
* <https://youtu.be/lTPT1cPfVmI?si=e2eBv2Y0vcXDh5fn>
* Стаття.

<https://www.geeksforgeeks.org/strings-in-cpp/>

* Що опрацьовано:
* Опрацьовано поняття символів і рядкових змінних, різницю між рядковими змінними в С і С++, корисні методи рядкових змінних у С++.
* Статус: Ознайомлений
* Початок опрацювання теми: 14.11.2023
* Звершення опрацювання теми: 15.12.2023
* Тема №3: Стандартна бібліотека
* Джерела Інформації:
* Відео.
* <https://youtu.be/Yevgn0yu5mI?si=CQVHOohlXNiGqKnM>
* <https://youtu.be/Yevgn0yu5mI?si=HKHqrdq9Uz54U_9I>
* Стаття.
* <https://www.geeksforgeeks.org/set-in-cpp-stl/>
* <https://www.geeksforgeeks.org/map-associative-containers-the-c-standard-template-library-stl/>
* Що опрацьовано:
* Опрацьовано складові стандартної бібліотеки – (вектори, списки, сети та мапи), алгоритми(сортування та пошуку), функції, ітератори.
* Статус: Ознайомлений
* Початок опрацювання теми: 14.12.2023
* Звершення опрацювання теми: 15.12.2023

**Виконання роботи:**

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

Завдання №1 **Algotester Lab 4**

* Варіант завдання:1
* Завдання №2 **Algotester Lab 6**
* Варіант завдання:1

Завдання №3 **Class Practice Work**

Завдання №4 **VNS Lab 6**

* Варіант завдання:14

Завдання №\_6 **VNS Lab 8**

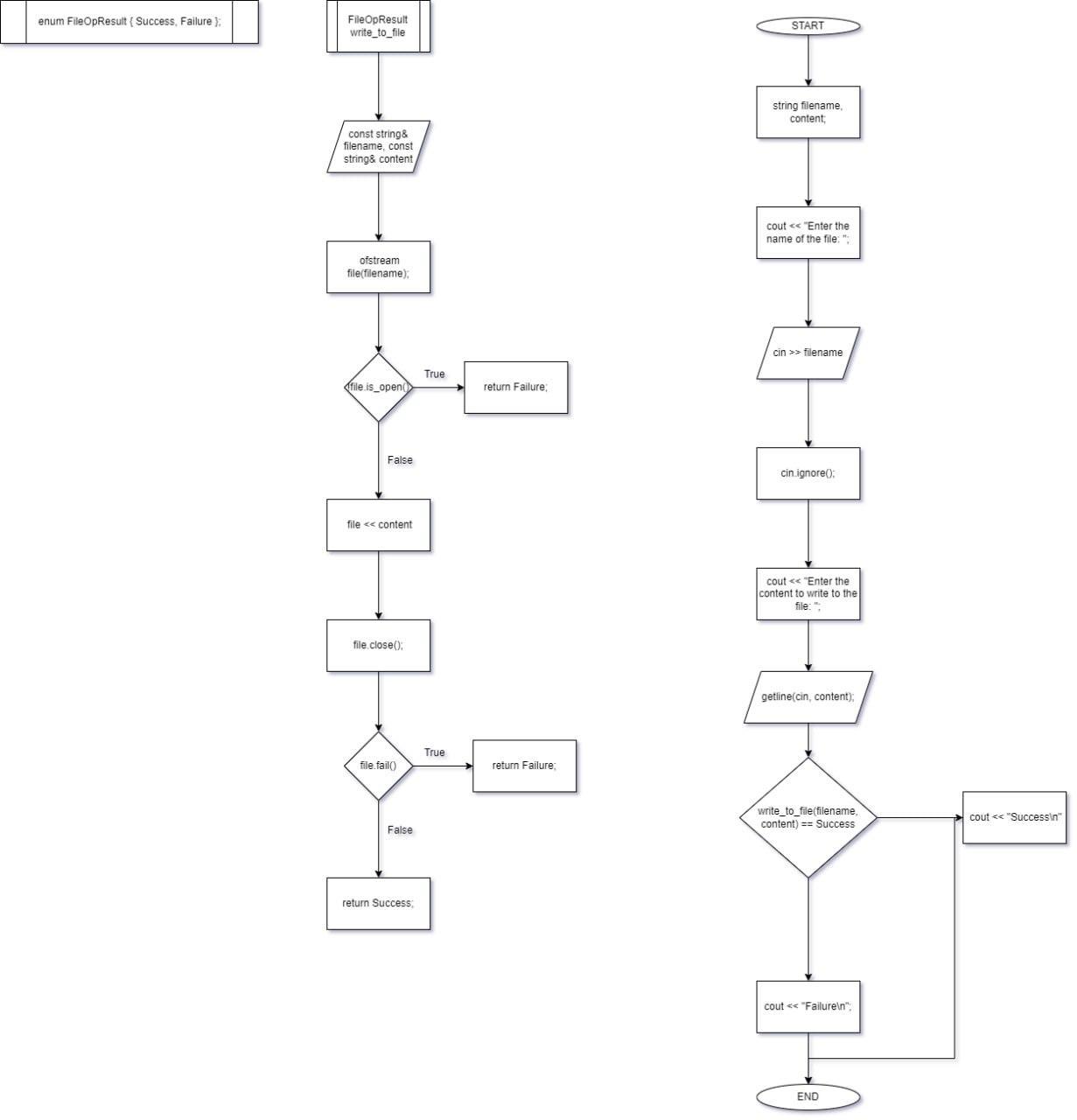
* Варіант завдання:14

Завдання №\_7 **VNS Lab 9**

* Варіант завдання:14

**2. Дизайн та планована оцінка часу виконання flow chart:**

Програма №1 Class Practice Work

* Блок-схема
* 
* Планований час на реалізацію:30 хв

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

[**https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/1037**](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/1037)

Завдання №1 **Algotester Lab 4**

[**https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/1037/files#diff-f3333263c352efc57caeefa0af47c27bf0ee026083c2a739e4e0b420a4879cda**](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/1037/files#diff-f3333263c352efc57caeefa0af47c27bf0ee026083c2a739e4e0b420a4879cda)

**#include <iostream>**

**#include <vector>**

**#include <algorithm>**

**using namespace std;**

**int main() {**

**// Зчитування розмірів та елементів першого вектора**

**int N;**

**cin >> N;**

**vector<int> vec1(N);**

**for (int i = 0; i < N; ++i) {**

**cin >> vec1[i];**

**}**

**// Зчитування розмірів та елементів другого вектора**

**int M;**

**cin >> M;**

**vector<int> vec2(M);**

**for (int i = 0; i < M; ++i) {**

**cin >> vec2[i];**

**}**

**// Сортування векторів для операцій, які вимагають відсортованих даних**

**sort(vec1.begin(), vec1.end());**

**sort(vec2.begin(), vec2.end());**

**// Виконання операцій над векторами**

**vector<int> diff1, diff2, intersection, unionVec, symmetricDifference;**

**set\_difference(vec1.begin(), vec1.end(), vec2.begin(), vec2.end(), back\_inserter(diff1));**

**set\_difference(vec2.begin(), vec2.end(), vec1.begin(), vec1.end(), back\_inserter(diff2));**

**set\_intersection(vec1.begin(), vec1.end(), vec2.begin(), vec2.end(), back\_inserter(intersection));**

**set\_union(vec1.begin(), vec1.end(), vec2.begin(), vec2.end(), back\_inserter(unionVec));**

**set\_symmetric\_difference(vec1.begin(), vec1.end(), vec2.begin(), vec2.end(), back\_inserter(symmetricDifference));**

**// Вивід результатів**

**cout << diff1.size() << endl;**

**for (int num : diff1) {**

**cout << num << " ";**

**}**

**cout << endl;**

**cout << diff2.size() << endl;**

**for (int num : diff2) {**

**cout << num << " ";**

**}**

**cout << endl;**

**cout << intersection.size() << endl;**

**for (int num : intersection) {**

**cout << num << " ";**

**}**

**cout << endl;**

**cout << unionVec.size() << endl;**

**for (int num : unionVec) {**

**cout << num << " ";**

**}**

**cout << endl;**

**cout << symmetricDifference.size() << endl;**

**for (int num : symmetricDifference) {**

**cout << num << " ";**

**}**

**cout << endl;**

**return 0;**

**}**

Завдання №2 **Algotester Lab 6**

[**https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/1037/files#diff-e7c24d2a915818e5df936454bdf21be4f72a00d4fca89dc27cbbfd946b91fa67**](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/1037/files#diff-e7c24d2a915818e5df936454bdf21be4f72a00d4fca89dc27cbbfd946b91fa67)

**#include <algorithm>**

**#include <iostream>**

**#include <string>**

**#include <unordered\_map>**

**#include <vector>**

**bool sort(char a, char b) {**

**return a > b;**

**}**

**int main() {**

**int N, K, m = 0;**

**std::cin >> N >> K;**

**std::string t, res;**

**std::vector<std::string> s;**

**std::unordered\_map<std::string, int> map;**

**for (int i = 0; i < N; i++) {**

**std::cin >> t;**

**std::transform(t.begin(), t.end(), t.begin(), ::tolower);**

**s.push\_back(t);**

**map[t]++;**

**}**

**for (int i = 0; i < N; i++) {**

**int count = map[s[i]];**

**if (count >= K) {**

**m += count;**

**res += s[i];**

**}**

**}**

**std::sort(res.begin(), res.end());**

**auto last = std::unique(res.begin(), res.end());**

**res.erase(last, res.end());**

**std::sort(res.begin(), res.end(), sort);**

**if (m >= K) {**

**std::cout << res.size() << std::endl;**

**for (auto ch : res) {**

**std::cout << ch << " ";**

**}**

**} else {**

**std::cout << "Empty!";**

**}**

**return 0;**

**}**

Завдання №3 **Class Practice Work**

[**https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/1037/files#diff-264353019c908d2a1b3fda7c0b56eea7ec7c158d598a6577bd3594e72def0ee6**](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/1037/files#diff-264353019c908d2a1b3fda7c0b56eea7ec7c158d598a6577bd3594e72def0ee6)

**#include <iostream>**

**#include <fstream>**

**#include <string>**

**using namespace std;**

**// Перерахування для результатів операцій з файлом**

**enum FileOpResult { Success, Failure };**

**// Функція для запису вмісту у файл**

**FileOpResult write\_to\_file(const string& filename, const string& content) {**

**// Відкриття файлу для запису**

**ofstream file(filename);**

**// Перевірка, чи файл успішно відкрито**

**if (!file.is\_open()) {**

**return Failure;**

**}**

**// Запис вмісту у файл**

**file << content;**

**// Закриття файлу**

**file.close();**

**// Перевірка, чи не виникли помилки під час запису**

**if (file.fail()) {**

**return Failure;**

**}**

**return Success;**

**}**

**int main() {**

**string filename, content;**

**// Введення ім'я файлу від користувача**

**cout << "Enter the name of the file: ";**

**cin >> filename;**

**// Ігнорування залишкових символів в буфері введення**

**cin.ignore();**

**// Введення вмісту файлу від користувача**

**cout << "Enter the content to write to the file: ";**

**getline(cin, content);**

**// Виклик функції для запису вмісту у файл і виведення результату**

**if (write\_to\_file(filename, content) == Success) {**

**cout << "Success\n";**

**} else {**

**cout << "Failure\n";**

**}**

**return 0;**

**}**

Завдання №4 **VNS Lab 6 - Task 14**

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/1037/files#diff-174a7f99e5e498e9a43ae5214984f180aa0f29ebc80ec81956e2a5a2cdb9f1d7>

#include <stdio.h>

#include <string.h>

#include <ctype.h>

void formattedString(char str[]) {

char result[256] = "";

char \*token = strtok(str, " ");

while (token != NULL) {

int il = 0, id = 0;

for (int i = 0; token[i] != '\0'; i++) {

if (isalpha(token[i])) {

il = 1;

} else if (isdigit(token[i])) {

id = 1;

}

}

if (il && id) {

strcat(result, token);

strcat(result, " ");

}

token = strtok(NULL, " ");

}

strcpy(str, result);

}

int main() {

char s[256];

printf("Enter a string: ");

gets(s);

printf("Original String: %s\n", s);

formattedString(s);

printf("Processed String: %s\n", s);

return 0;

}

Завдання №5 **VNS Lab 8 - Task 14**

[**https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/1037/files#diff-0cafa6c7b777599930f84cfd9b2cf496cb395deefa76a20509ed6b51eb59c189**](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/1037/files#diff-0cafa6c7b777599930f84cfd9b2cf496cb395deefa76a20509ed6b51eb59c189)

**#include <iostream>**

**#include <fstream>**

**#include <vector>**

**#include <cstdlib>**

**using namespace std;**

**// Створення структури Стадіон**

**struct Stadium {**

**string name;**

**string address;**

**int capacity;**

**string sports;**

**// Структура за замовчанням**

**Stadium() : name(""), address(""), capacity(0), sports("") {}**

**// структура**

**Stadium(const string& n, const string& a, int c, const string& s)**

**: name(n), address(a), capacity(c), sports(s) {}**

**// вивід інформації про Стадіон**

**void display() const {**

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

**cout << "Address: " << address << endl;**

**cout << "Capacity: " << capacity << endl;**

**cout << "Sports: " << sports << endl << endl;**

**}**

**};**

**// функція для зміни структури**

**void displayFile(const string& filename);**

**void deleteStadium(const string& filename, const string& nameToDelete);**

**void addStadium(const string& filename, int position, const Stadium& stadium1, const Stadium& stadium2);**

**int main() {**

**const string filename = "stadiums.dat";**

**// заповнення файлу данними**

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

**Stadium stadium1("Stadium1", "Address1", 10000, "Football");**

**Stadium stadium2("Stadium2", "Address2", 15000, "Hockey");**

**Stadium stadium3("Stadium3", "Address3", 20000, "Basketball");**

**initialFile.write(reinterpret\_cast<const char\*>(&stadium1), sizeof(Stadium));**

**initialFile.write(reinterpret\_cast<const char\*>(&stadium2), sizeof(Stadium));**

**initialFile.write(reinterpret\_cast<const char\*>(&stadium3), sizeof(Stadium));**

**initialFile.close();**

**cout << "Initial state of the file:" << endl;**

**displayFile(filename);**

**cout << "Initial state of the file:" << endl;**

**displayFile(filename);**

**// видалення елементу з заданою назвою (з умови)**

**string stadiumToDelete = "Stadium2";**

**deleteStadium(filename, stadiumToDelete);**

**cout << "\nAfter deleting the stadium \"" << stadiumToDelete << "\":" << endl;**

**displayFile(filename);**

**// добавлення двох елементів після зазначеного номеру**

**int positionToAdd = 1; // add after the first element**

**Stadium newStadium1("New Stadium1", "New Address1", 12000, "Tennis");**

**Stadium newStadium2("New Stadium2", "New Address2", 18000, "Volleyball");**

**addStadium(filename, positionToAdd, newStadium1, newStadium2);**

**cout << "\nAfter adding new stadiums:" << endl;**

**displayFile(filename);**

**return 0;**

**}**

**// функція для видалення елемента із заданою назвою**

**void deleteStadium(const string& filename, const string& nameToDelete) {**

**ifstream fileIn(filename, ios::binary);**

**ofstream fileOut("temp.dat", ios::binary);**

**if (!fileIn || !fileOut) {**

**cerr << "Error opening the file." << endl;**

**exit(1);**

**}**

**Stadium tempStadium;**

**while (fileIn.read(reinterpret\_cast<char\*>(&tempStadium), sizeof(Stadium))) {**

**if (tempStadium.name != nameToDelete) {**

**fileOut.write(reinterpret\_cast<const char\*>(&tempStadium), sizeof(Stadium));**

**}**

**}**

**fileIn.close();**

**fileOut.close();**

**remove(filename.c\_str());**

**rename("temp.dat", filename.c\_str());**

**}**

**void addStadium(const string& filename, int position, const Stadium& stadium1, const Stadium& stadium2) {**

**ifstream fileIn(filename, ios::binary);**

**ofstream fileOut("temp.dat", ios::binary);**

**if (!fileIn || !fileOut) {**

**cerr << "Error opening the file." << endl;**

**exit(1);**

**}**

**Stadium tempStadium;**

**int currentPosition = 1;**

**while (fileIn.read(reinterpret\_cast<char\*>(&tempStadium), sizeof(Stadium))) {**

**fileOut.write(reinterpret\_cast<const char\*>(&tempStadium), sizeof(Stadium));**

**if (currentPosition == position) {**

**fileOut.write(reinterpret\_cast<const char\*>(&stadium1), sizeof(Stadium));**

**fileOut.write(reinterpret\_cast<const char\*>(&stadium2), sizeof(Stadium));**

**}**

**currentPosition++;**

**}**

**fileIn.close();**

**fileOut.close();**

**remove(filename.c\_str());**

**rename("temp.dat", filename.c\_str());**

**}**

**void displayFile(const string& filename) {**

**ifstream fileIn(filename, ios::binary);**

**if (!fileIn) {**

**cerr << "Error opening the file." << endl;**

**return;**

**}**

**Stadium tempStadium;**

**while (fileIn.read(reinterpret\_cast<char\*>(&tempStadium), sizeof(Stadium))) {**

**tempStadium.display();**

**}**

**fileIn.close();**

**}**

Завдання №\_7 **VNS Lab 9 - Task 14**

[**https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/1037/files#diff-0cf45bdae10a69af2a0b053caf99cb9b52d9db995176d882688708c4c34cfbda**](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/1037/files#diff-0cf45bdae10a69af2a0b053caf99cb9b52d9db995176d882688708c4c34cfbda)

**#include <iostream>**

**#include <fstream>**

**#include <string>**

**#include <algorithm>**

**using namespace std;**

**// Функція для підрахунку голосних букв у рядку**

**int countVowels(const string& line) {**

**const string vowels = "aeiouAEIOU";**

**return count\_if(line.begin(), line.end(), [&vowels](char c) {**

**return find(vowels.begin(), vowels.end(), c) != vowels.end();**

**});**

**}**

**int main() {**

**// Відкритя файлів для читання та запису**

**ifstream inputFile("F1.txt");**

**ofstream outputFile("F2.txt");**

**// Задаємо номери рядків N1 та N2**

**int N1 = 3;**

**int N2 = 7;**

**// Перевірка на помилку, (чи файли вдалося відкрити)**

**if (!inputFile) {**

**cerr << "ERROR 404\_1" << endl;**

**return 1;**

**}**

**if (!outputFile) {**

**cerr << "ERROR 404\_2" << endl;**

**return 1;**

**}**

**string line;**

**int maxVowelsLine = -1;**

**int maxVowelsCount = 0;**

**// Читаємо рядки з файлу F1**

**for (int i = 1; i <= N2 && getline(inputFile, line); ++i) {**

**// Перевіряємо, чи поточний рядок входить в діапазон N1-N2 та не містить букву 'A'**

**if (i >= N1 && i <= N2 && line.find('A') == string::npos) {**

**// Записуємо відповідні рядки у файл F2**

**outputFile << line << endl;**

**// Знаходимо рядок з найбільшою кількістю голосних букв**

**int vowelsCount = countVowels(line);**

**if (vowelsCount > maxVowelsCount) {**

**maxVowelsCount = vowelsCount;**

**maxVowelsLine = i;**

**}**

**}**

**}**

**// Закриваємо файли**

**inputFile.close();**

**outputFile.close();**

**// Виводимо результат**

**if (maxVowelsLine != -1) {**

**cout << maxVowelsLine << endl;**

**} else {**

**cout << "there's no lines that i need"<< endl;**

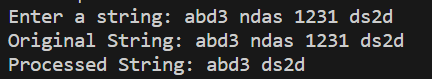
**}**

**return 0;**

**}**

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

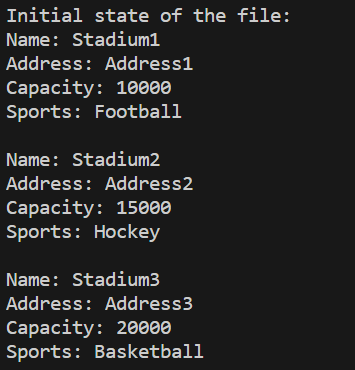
Завдання №1 **VNS Lab 6 - Task 1**

****

*Figure 2.*

Час затрачений на виконання завдання: 1 година

Завдання №\_2 **VNS Lab 8 - Task 1**

****

*Figure 3.*

Час затрачений на виконання завдання: 1 година

Завдання №\_3 **VNS Lab 9 - Task 1**



*Figure 4.*

Час затрачений на виконання завдання: 1 година

Завдання №4 **Algotester Lab 4**

[**https://algotester.com/en/ProblemSolution/Display/1525450**](https://algotester.com/en/ProblemSolution/Display/1525450)

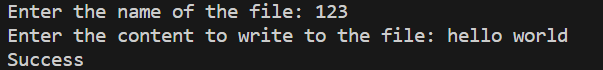
Час затрачений на виконання завдання: 1 година

Завдання №5 **Algotester Lab 6**

[**https://algotester.com/en/ProblemSolution/Display/1525648**](https://algotester.com/en/ProblemSolution/Display/1525648)

Час затрачений на виконання завдання: 1 година

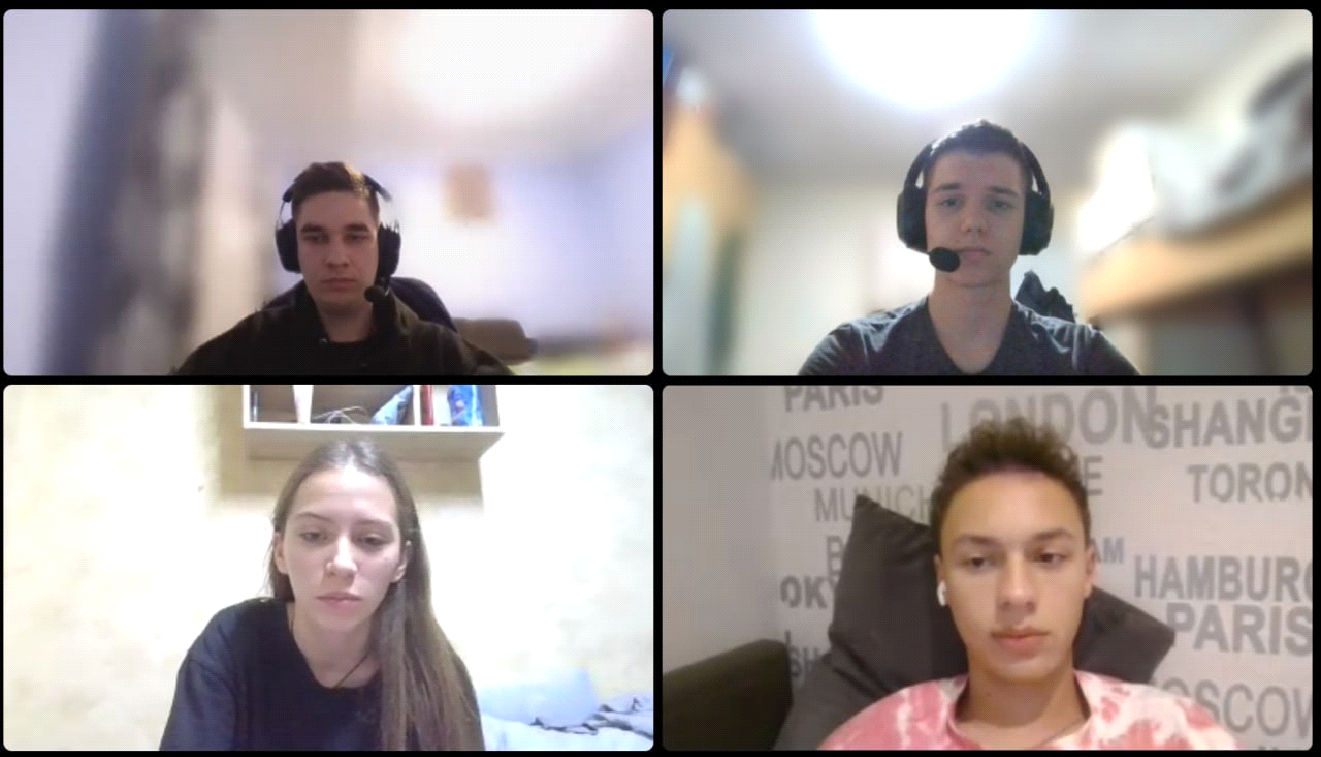
Завдання №6 **Class Practice Work**

****

*Figure 5.*

Час затрачений на виконання завдання: 1 година

**5. Кооперація з командою:**



*Figure 6.*

**Висновки:**

Під час виконання епіку 5 я ознайомився з бінарними та текстовими файлами, символьними та рядковими змінними, стандартною бібліотекою та системами числення.