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-14

Мруць Остап Мар’янович

# **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:**

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Робота з [файл](https://vns.lpnu.ua/mod/glossary/showentry.php?eid=210916&displayformat=dictionary)ами
* Тема №2: Динамічні [с](https://vns.lpnu.ua/mod/glossary/showentry.php?eid=209449&displayformat=dictionary)труктури даних
* Тема №3: Бібліотеки

1. Індивідуальний план опрацювання теорії:

* Тема №1: Робота з [файл](https://vns.lpnu.ua/mod/glossary/showentry.php?eid=210916&displayformat=dictionary)ами
  + Джерела Інформації
    - Лекції
    - Лекційний матеріал на ВНС
  + Що опрацьовано:
    - О[с](https://vns.lpnu.ua/mod/glossary/showentry.php?eid=209449&displayformat=dictionary)нови роботи з [файл](https://vns.lpnu.ua/mod/glossary/showentry.php?eid=210916&displayformat=dictionary)ами.
    - Ввід і вивід [с](https://vns.lpnu.ua/mod/glossary/showentry.php?eid=209449&displayformat=dictionary)труктурованих [файл](https://vns.lpnu.ua/mod/glossary/showentry.php?eid=210916&displayformat=dictionary)ів.
    - Потоковий ввід-вивід. [Відкриття й закриття потоку](https://vns.lpnu.ua/mod/glossary/showentry.php?eid=210920&displayformat=dictionary). Блоковий ввід-вивід.
    - Робота з тек[с](https://vns.lpnu.ua/mod/glossary/showentry.php?eid=209449&displayformat=dictionary)товими [файл](https://vns.lpnu.ua/mod/glossary/showentry.php?eid=210916&displayformat=dictionary)ами. О[с](https://vns.lpnu.ua/mod/glossary/showentry.php?eid=209449&displayformat=dictionary)новні функції [файл](https://vns.lpnu.ua/mod/glossary/showentry.php?eid=210916&displayformat=dictionary)ової [с](https://vns.lpnu.ua/mod/glossary/showentry.php?eid=209449&displayformat=dictionary)и[с](https://vns.lpnu.ua/mod/glossary/showentry.php?eid=209449&displayformat=dictionary)теми.
  + Статус: Ознайомлений
* Тема №2: Динамічні [с](https://vns.lpnu.ua/mod/glossary/showentry.php?eid=209449&displayformat=dictionary)труктури даних
  + Джерела Інформації:
    - Лекції
    - Лекційний матеріал на ВНС
  + Що опрацьовано:
    - [Лінійний однонаправлений список](https://vns.lpnu.ua/mod/glossary/showentry.php?eid=210912&displayformat=dictionary).
    - Робота із двонаправленими [с](https://vns.lpnu.ua/mod/glossary/showentry.php?eid=209449&displayformat=dictionary)пи[с](https://vns.lpnu.ua/mod/glossary/showentry.php?eid=209449&displayformat=dictionary)ками.
  + Статус: Ознайомлений
* Тема №3: Бібліотеки
  + Джерела Інформації:
    - Лекції
    - Лекційний матеріал на ВНС
  + Що опрацьовано:
    - Стандартна бібліотека та деталі/методи роботи з файлами.
    - Створення й використання бібліотек.
  + Статус: Ознайомлений

# **Виконання роботи:**

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

Завдання №1 - Theory Education Activities

* Вивчення теорії

Завдання №2 - Requirements management (understand tasks) and design activities (draw flow diagrams and estimate tasks 3-9)

* Створення діаграм для програм в draw io

Завдання №3 - Lab# programming: VNS Lab 6

* Варіант № 19
* Деталі завдання: Задано рядок, що складається із символів. Символи поєднуються в слова. Слова одне від одного відокремлюються одним або декількома пробілами. Наприкінці тексту ставиться крапка. Текст містить не більше 255 символів. Виконати ввід рядка, використовуючи функцію gets(s) і здійснити обробку рядка у відповідності зі своїм варіантом. Надрукувати всі слова, які співпадають з її першим словом.
* Важливі деталі для врахування в імплементації програми: Кожне слово відокремлюється пробілом, що допоможе легше порівнювати слова, також в кінці ставиться крапка, що допоможе зрозуміти коли стрічка закінчиться.
* Знищити з рядка всі слова, які не є ідентифікаторами.

Завдання №4 - Lab# programming: VNS Lab 8

* Варіант № 19
* 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. Передбачити повідомлення про помилки при відкритті файлу й виконанні операцій вводу/виводу.
* Структура "Фільм":
* - назва;
* - режисер;
* - країна;
* - прибуток.
* Знищити 2 елементи з кінця файлу, додати елемент після елемента із
* зазначеною назвою.
* Важливі деталі для врахування в імплементації програми: Всі дії над структурою потрібно здійснювати через відповідну функцію, також при помилці відкриття файлу.

Завдання №5 - Lab# programming: VNS Lab 9

* Варіант № 19
* 1) Скопіювати з файлу F1 у файл F2 всі рядки, у яких немає слів, що збігаються
* з першим словом.
* 2) Визначити кількість приголосних букв у першому рядку файлу F2.

Завдання №6 - Lab# programming: Algotester Lab 4

* Варіант № 3
* Важливі деталі для врахування в імплементації програми: **Пам’ятайте, ви маєте написати 2 варіанти розвязку, один з використанням засобів STL (std::unique, std::sort, std::rotate), інший зі своєю реалізацією**

Завдання №7 - Lab# programming: Algotester Lab 6

* Варіант № 3
* Важливі деталі для врахування в імплементації програми: Пам’ятайте, що не можна вписувати цифру більшу, ніж розмір сітки. (Тобто в сітку 3×3 не можна вписати цифру 4)

Завдання №8 - Practice# programming: Class Practice Task

* Деталі завдання: У вас Реалізувати функцію створення файлу 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 файлу.

* Важливі деталі для врахування в імплементації програми: Статус операції має проходити через enum. При створенні файлу з ім`ям вже існуючого файлу потрібно переписати його з новим, вписаним користувачем, змістом.

Завдання №9 - Practice# programming: Self Practice Task

* Деталі завдання: <https://algotester.com/uk/ContestProblem/DisplayWithEditor/134641>
* Важливі деталі для врахування в імплементації програми: 1≤N,M≤103, 1≤x≤N, 1≤y≤M, спочатку варто знайти число найвищої точки, після цього згенерувати гору буде простіше

Завдання №10 - Result Documentation Report and Outcomes Placement Activities (Docs and Programs on GitHub)

* Створення звіту і завантажити його з програмами на GitHub

Завдання №11 - Results Evaluation and Release

* Захист звіту.

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

Програма №1 VNS Lab 6

* Важливі деталі для врахування в імплементації програми: Були використані функції strtok\_s для розбиття слів на токени для більш зручної перевірки, а також strcmp для самої перевірки
* Важливі деталі для врахування в імплементації програми: В програмі досить багато функцій, важливо коректно передавати дані через них. А також потрібно збільшувати кількість персон лише при їх добавленні

Програма №3 VNS Lab 9

* Важливі деталі для врахування в імплементації програми: sizeof повертає кількість бітів, що добре підходить для виведення рядка через getline, коли ж для підрахування кількості букв краще підходить length()

Програма №4 Algotester Lab 4

* Важливі деталі для врахування в імплементації програми: sizeof відображає кількість байтів, це не допоможе в вимірі довжини рядка. В довжині рядку приймає участь і його кінець, тобто місце під новий рядок.

Програма №5 Algotester Lab 6

* Важливі деталі для врахування в імплементації програми: Якщо координати за межами масиву, то можна одразу вивести «О», так само якщо координати вказані на якійсь фігурі, то можна одразу вивести «Х». Якщо багато фігур стають в одну незайняту клітинку, то пітрібно вивести їх в алфавітному порядку і без повторень

Програма №6 Class Practice Task

* Важливі деталі для врахування в імплементації програми: Для зчитування рядка використовується функція getline(), а для перевірки чи файл відкрився is\_open().

Програма №7 Self Practice Task

* Важливі деталі для врахування в імплементації програми: Щоб виміряти цифру якоїсь частини гори потрібно спершу знайти яке число буде на найбільшій вершині гори

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

Завдання №1: VNS Lab 6

#include <iostream>

#include <sstream>

#include <string>

#include <cctype>

bool isIdentifier(const std::string& word) {

if (word.empty() || !isalpha(word[0]) && word[0] != '\_') {

return false;

}

for (size\_t i = 1; i < word.length(); ++i) {

if (!isalnum(word[i]) && word[i] != '\_') {

return false;

}

}

return true;

}

int main() {

std::istringstream iss("slovo \_slovo\_ slovechko-s slovo123 \_slovo 1slovechko");

std::string word;

bool isFirstWord = true;

while (iss >> word) {

if (isIdentifier(word)) {

if (!isFirstWord) {

std::cout << ' ';

}

std::cout << word;

isFirstWord = false;

}

}

std::cout << "." << std::endl;

return 0;

}

*Рисунок 2: Код завдання з VNS №6*

Завдання №2: VNS Lab 8

#include <iostream>

#include <fstream>

#include <string>

#include <vector>

int main() {

std::string file\_path = "C:\\Users\\MAHINA\\Desktop\\MyE5\\t.txt";

std::fstream file(file\_path, std::ios::in | std::ios::out);

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

std::cout << "Не вдалося відкрити файл!" << std::endl;

return 1;

}

std::vector<std::string> lines;

std::string line;

while (std::getline(file, line)) {

lines.push\_back(line);

}

std::string country, profit;

std::cout << "Країна: ";

std::getline(std::cin, country);

std::cout << "Прибуток: ";

std::getline(std::cin, profit);

lines[lines.size() - 2] = country;

lines.back() = profit;

file.clear();

file.seekp(0, std::ios::beg);

for (const auto &l : lines) {

file << l << std::endl;

}

std::cout << "Зміни внесено" << std::endl;

file.close();

return 0;

}

*Рисунок 3: Код завдання з VNS №8*

Завдання №3: VNS Lab 9

#include <iostream>

#include <fstream>

#include <vector>

#include <algorithm>

bool isVowel(char c) {

c = std::tolower(c);

return (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u');

}

int countConsonants(const std::string& word) {

int consonantCount = 0;

for (char c : word) {

if (std::isalpha(c) && !isVowel(c)) {

consonantCount++;

}

}

return consonantCount;

}

int main() {

std::ofstream file1("C:\\Users\\MAHINA\\Desktop\\MyE5\\F1.txt");

if (!file1.is\_open()) {

std::cerr << "Unable to open file for writing.\n";

return 1;

}

std::vector<std::string> words;

std::cout << "Enter 10 words:\n";

for (int i = 0; i < 10; ++i) {

std::string word;

std::cin >> word;

file1 << word << "\n";

words.push\_back(word);

}

file1.close();

std::ifstream file2("C:\\Users\\MAHINA\\Desktop\\MyE5\\F1.txt");

if (!file2.is\_open()) {

std::cerr << "Unable to open file for reading.\n";

return 1;

}

std::string firstWord;

file2 >> firstWord;

file2.close();

int consonantCount = countConsonants(firstWord);

std::cout << "Consonant count in the first word (" << firstWord << "): " << consonantCount << "\n";

std::ofstream file3("C:\\Users\\MAHINA\\Desktop\\MyE5\\F2.txt");

if (!file3.is\_open()) {

std::cerr << "Unable to open file for writing.\n";

return 1;

}

bool foundDuplicate = false;

for (const std::string& word : words) {

if (word == firstWord) {

if (!foundDuplicate) {

foundDuplicate = true;

continue;

}

}

file3 << word << "\n";

}

file3.close();

return 0;

}

*Рисунок 4: Код завдання з VNS №9*

Завдання №4: Algotester Lab 4

Метод 2:

##include <stdio.h>

#include <vector>

#include <algorithm>

int main() {

int totalNumbers;

scanf("%d", &totalNumbers);

std::vector<int> inputNumbers(totalNumbers);

for (int i = 0; i < totalNumbers; ++i) {

scanf("%d", &inputNumbers[i]);

}

std::vector<int> sortedNumbers;

int countDivisible = 0;

for (int i = 0; i < totalNumbers; ++i) {

sortedNumbers.push\_back(inputNumbers[i]);

if (inputNumbers[i] % 3 == 0) {

countDivisible++;

}

}

std::sort(sortedNumbers.begin(), sortedNumbers.end(), [countDivisible](int a, int b) {

if (a % 3 == 1 && b % 3 == 1) {

return a > b;

}

return std::make\_pair(a % 3, a) < std::make\_pair(b % 3, b);

});

sortedNumbers.erase(std::unique(sortedNumbers.begin(), sortedNumbers.end()), sortedNumbers.end());

printf("%zu\n", sortedNumbers.size());

for (int num : sortedNumbers) {

printf("%d ", num);

}

return 0;

}

*Рисунок 5: Код завдання з Algotester №6*

*##include <iostream>*

*#include <vector>*

*#include <algorithm>*

*struct Coordinate {*

*int x;*

*int y;*

*};*

*std::vector<int> getPossibleValues(const std::vector<std::vector<int>>& sudoku, int row, int col) {*

*int N = sudoku.size();*

*std::vector<int> possible;*

*for (int value = 1; value <= N; ++value) {*

*possible.push\_back(value);*

*}*

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

*auto it = std::find(possible.begin(), possible.end(), sudoku[row][j]);*

*if (it != possible.end()) {*

*possible.erase(it);*

*}*

*}*

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

*auto it = std::find(possible.begin(), possible.end(), sudoku[i][col]);*

*if (it != possible.end()) {*

*possible.erase(it);*

*}*

*}*

*return possible;*

*}*

*int main() {*

*int size;*

*int queries;*

*std::cin >> size;*

*std::vector<std::vector<int>> sudoku(size, std::vector<int>(size, 0));*

*std::vector<int> numbers;*

*std::vector<Coordinate> coordinates;*

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

*int a;*

*std::cin >> a;*

*numbers.push\_back(a);*

*}*

*std::cin >> queries;*

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

*int x, y;*

*std::cin >> x >> y;*

*coordinates.push\_back({ x, y });*

*}*

*for (int i = 0; i < numbers.size(); ++i) {*

*int number = numbers[i];*

*for (int j = size - 1; j >= 0; --j) {*

*sudoku[i][j] = number % 10;*

*number /= 10;*

*}*

*}*

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

*if (sudoku[coordinates[i].x - 1][coordinates[i].y - 1] != 0) {*

*std::cout << 1 << '\n';*

*std::cout << sudoku[coordinates[i].x - 1][coordinates[i].y - 1] << '\n' << '\n';*

*} else {*

*std::vector<int> possibleValues = getPossibleValues(sudoku, coordinates[i].x - 1, coordinates[i].y - 1);*

*std::cout << possibleValues.size() << '\n';*

*for (int j = 0; j < possibleValues.size(); j++) {*

*std::cout << possibleValues[j] << " ";*

*}*

*std::cout << '\n' << '\n';*

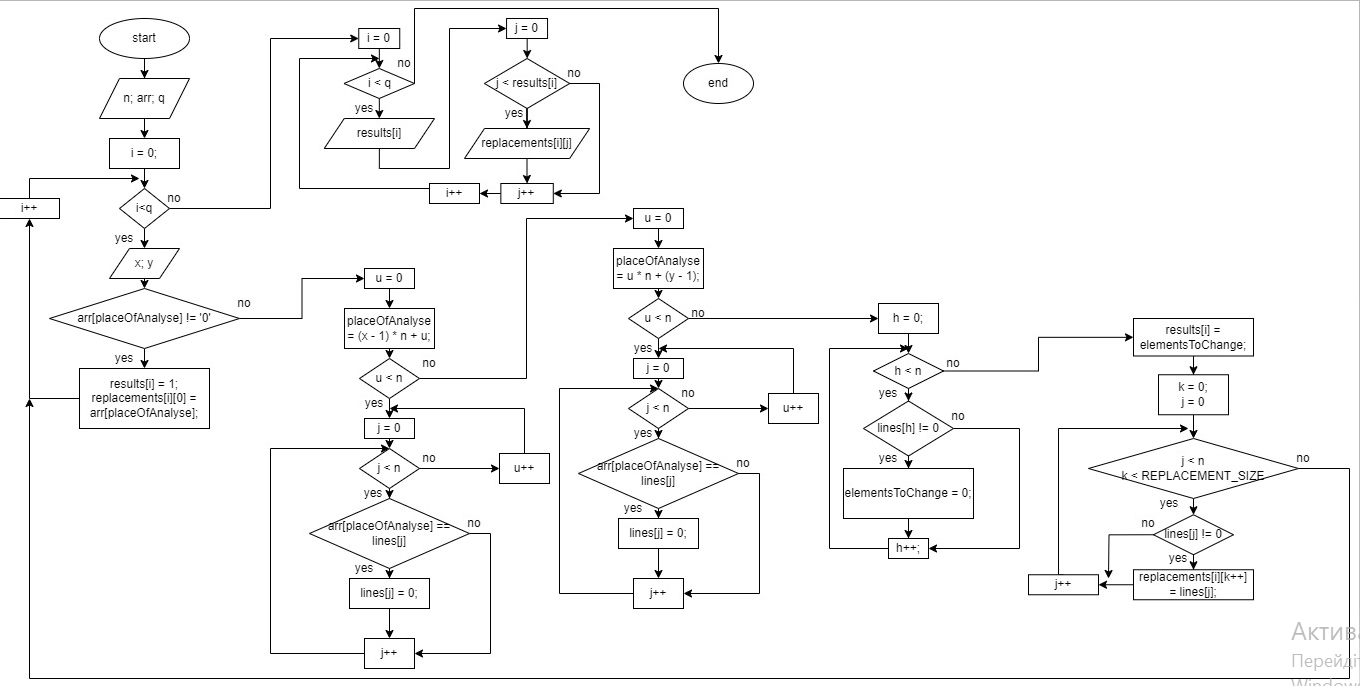
*}*

*}*

*return 0;*

*}*

*Рисунок 6: Код завдання з Algotester №6*

**

*Рисунок 7: Блок схема до програми №2*

Завдання №6: Class Practice Task

##include <iostream>

#include <fstream>

#include <string>

bool write\_to\_file(const std::string& filename) {

std::ofstream file(filename);

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

std::cout << "Помилка відкриття файлу для запису.\n";

return false;

}

std::string input\_text;

std::cout << "Введіть текст для запису в файл " << filename << ":\n";

std::getline(std::cin >> std::ws, input\_text);

file << input\_text;

file.close();

std::cout << "Текст успішно записано у файл " << filename << ".\n";

return true;

}

bool copy\_file(const std::string& source\_filename, const std::string& destination\_filename) {

std::ifstream source\_file(source\_filename);

if (!source\_file.is\_open()) {

std::cout << "Помилка відкриття файлу для читання.\n";

return false;

}

std::ofstream destination\_file(destination\_filename);

if (!destination\_file.is\_open()) {

std::cout << "Помилка відкриття/створення файлу для запису.\n";

return false;

}

destination\_file << source\_file.rdbuf();

source\_file.close();

destination\_file.close();

std::cout << "Вміст файлу " << source\_filename << " успішно скопійовано в файл " << destination\_filename << ".\n";

return true;

}

int main() {

const std::string filename = "C:\\Users\\MAHINA\\Desktop\\MyE5\\example.txt";

std::string copy\_destination\_filename;

if (write\_to\_file(filename)) {

std::cout << "Введіть назву файлу, куди скопіювати вміст файлу " << filename << ":\n";

std::cin >> copy\_destination\_filename;

if (copy\_file(filename, copy\_destination\_filename)) {

} else {

std::cout << "Не вдалося скопіювати вміст файлу.\n";

}

} else {

std::cout << "Не вдалося записати у файл " << filename << ".\n";

}

return 0;

}

*Рисунок 7: Код програми з практичної роботи*

Завдання №7: Self Practice Task

##include <iostream>

#include <fstream>

#include <string>

#include <vector>

int main() {

std::vector<std::string> words = {

"Definitely", "Separate", "Occurred", "Accommodate", "Embarrass",

"Receive", "Weird", "Occasion", "Privilege", "Conscious"

};

std::ofstream file("C:\\Users\\MAHINA\\Desktop\\MyE5\\words.txt");

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

std::cerr << "Unable to open the file." << std::endl;

return 1;

}

bool incorrectWritten = false;

file << "Correct words:" << std::endl;

for (const std::string& word : words) {

std::string userInput;

std::cout << "Please enter the word '" << word << "': ";

std::cin >> userInput;

if (userInput == word) {

file << word << std::endl;

} else {

if (!incorrectWritten) {

file << std::endl;

file << "UnCorrect words:" << std::endl;

file << std::endl;

incorrectWritten = true;

}

file << userInput << std::endl;

}

}

file.close();

std::cout << "Check the file for the results." << std::endl;

return 0;

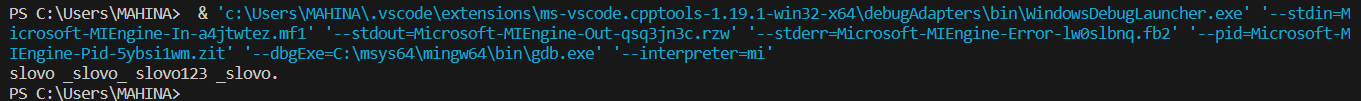
}

*Рисунок 8: Код програми для самостійного опрацювання*

## **4. Результати виконання завдань:**

Завдання №1: VNS Lab 6

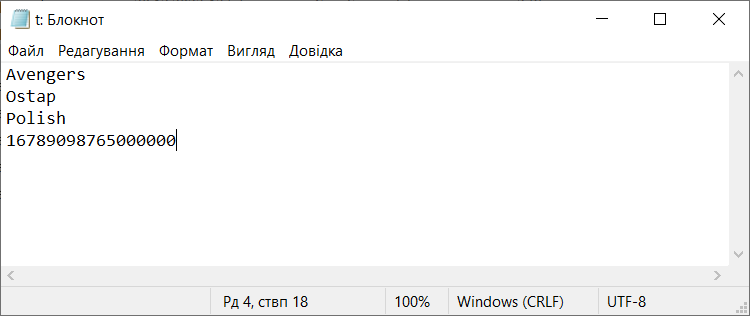
Результати виконання програми:

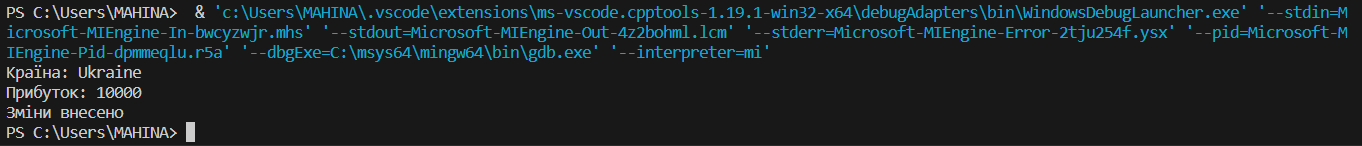


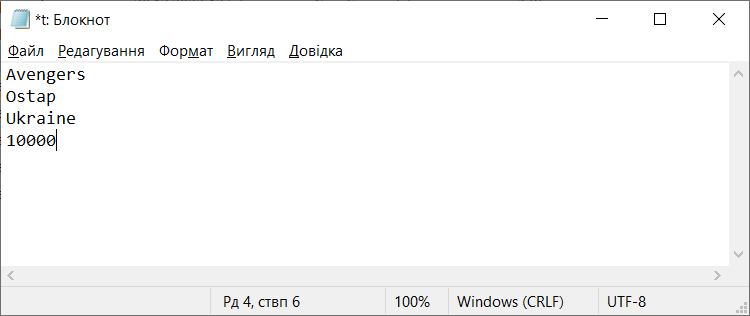
*Рисунок 9: Результати виконання програми з VNS №6*

Завдання №2: VNS Lab 8

Результати виконання програми:



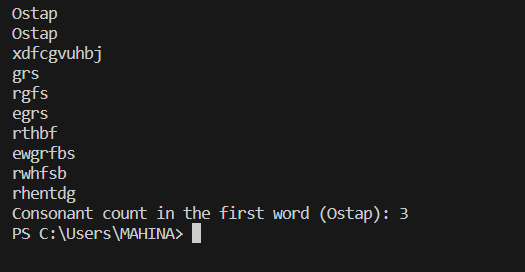


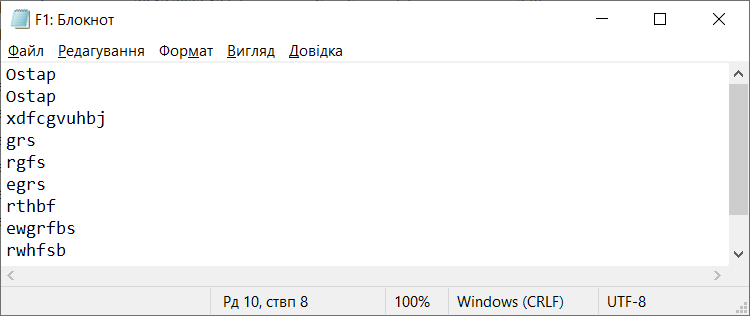


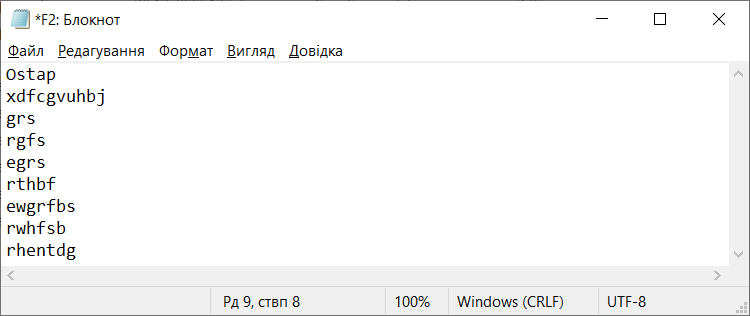
*Рисунок 10: Результати виконання програми з VNS №8*

Час затрачений на виконання завдання: 3 години

Завдання №3: VNS Lab 9



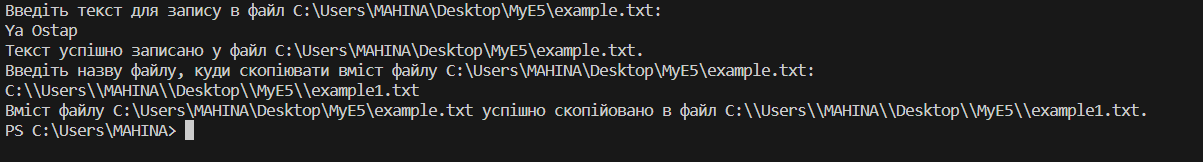




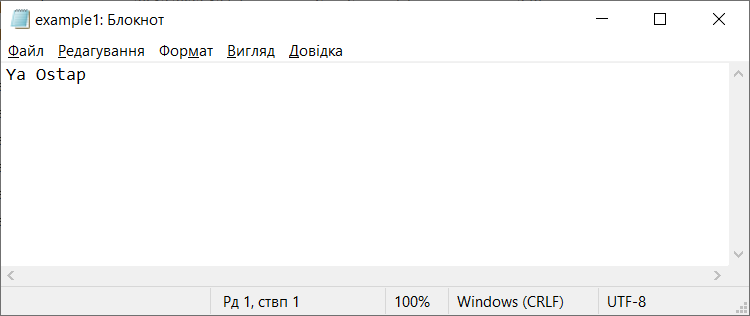
*Рисунок 11: Результати виконання програми з VNS №9*

Завдання №6: Class Practice Task

Результати виконання програми:



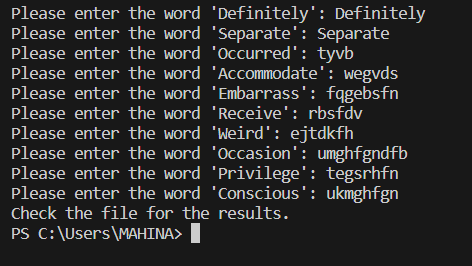
*Рисунок 14: Результати виконання практичної роботи*

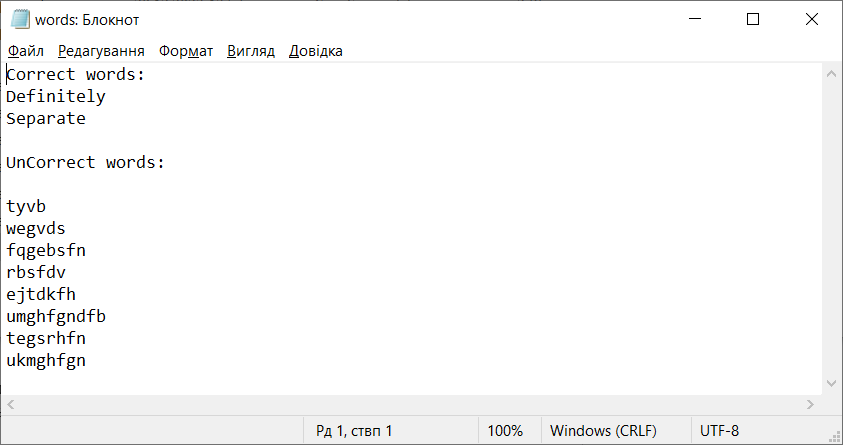


*Рисунок 15: Результати виконання практичної роботи в текстовому документі*

Завдання №7: Self Practice Task

Результати виконання програми:





*Рисунок 16: Результати виконання роботи для самостійного опрацювання*

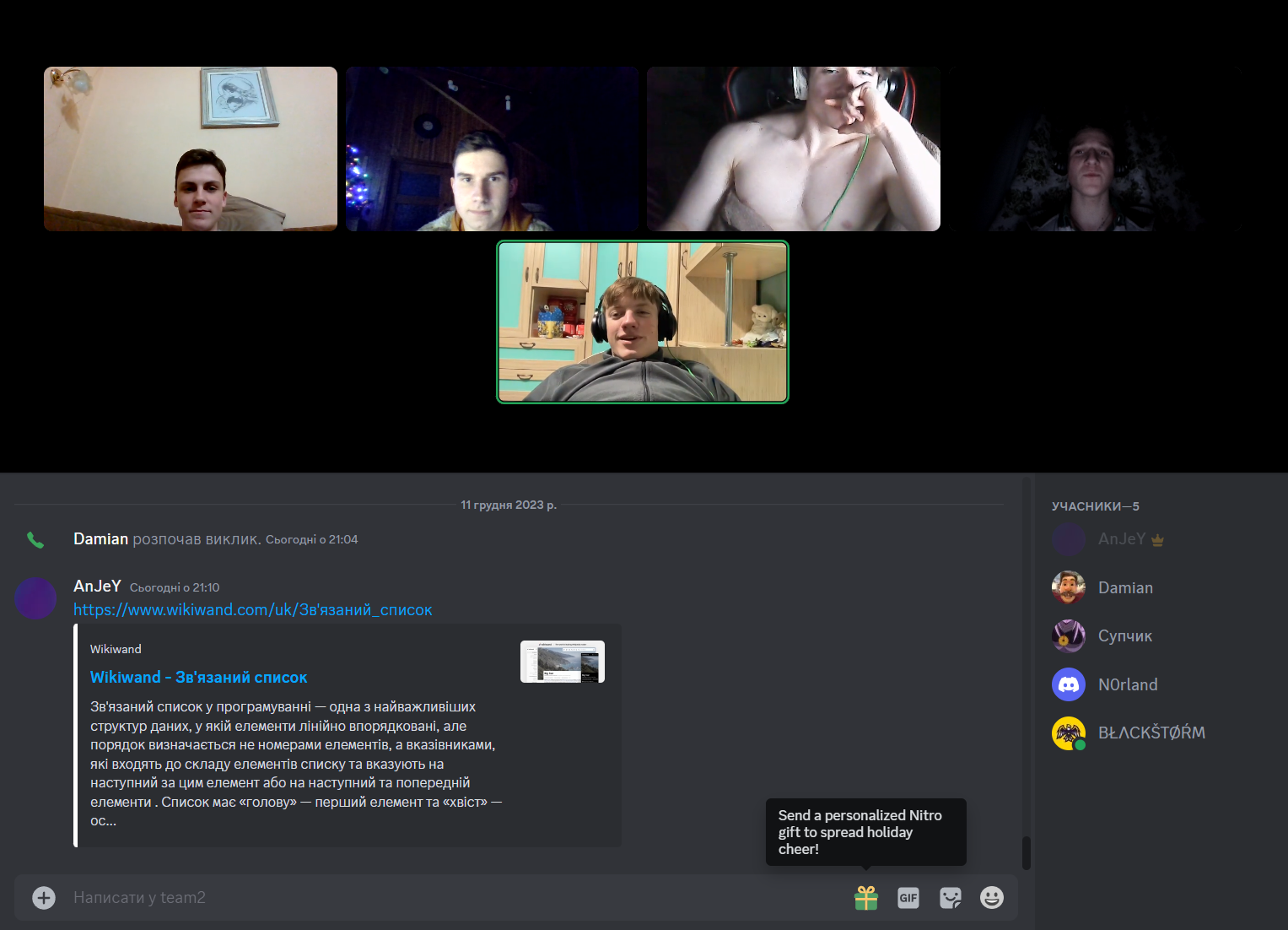
Деталі по виконанню:

* Перше завдяки модулю числа було знайдено число максимальної вершини і виставлено кожній вершині протилежне значення (максимальна вершина = 0), після чого кожну вершину відняв від максимального значення по модулю, що зробило значення коректними

Час затрачений на виконання завдання: 1 година

## **5. Кооперація з командою:**

Знімки конференції:



*Рисунок 17: Знімок зустрічі з командою*

# **Висновки:**

# Протягом лабораторних та практичних робіт №5було вивчено як створювати файли, повторено різні системи числення, також було вивчено як вписувати і зчитувати інформацію файлу, створено й використано бібліотеки. Були онлайн зустрічі з командою, в яких ми владнали всі непорозуміння.