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

Сокаль Олег Ігорович

# **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
  + Початок опрацювання теми: 26.11.2023
  + Звершення опрацювання теми: 27.11.2023
* 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
  + Початок опрацювання теми: 27.11.2023
  + Звершення опрацювання теми: 28.11.2023
* 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
  + Початок опрацювання теми: 28.11.2023
  + Звершення опрацювання теми: 29.11.2023

# **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

* Час на виконання: 2 години
* 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
* Час на виконання: 3 години
* 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

* Час на виконання: 1 година
* 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

* Час на виконання: 1 година і 30 хвилин
* 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

* Час на виконання: 6 годин
* 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

* Час на виконання: 1 година
* 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

* Час на виконання: 2 години
* 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 <stdio.h>

#include <string.h>

#include <ctype.h>

void identificatorOrNot(char str[]) {

char \*token;

token = strtok(str, " ");

while (token != NULL) {

int isValid = 1;

for (int i = 0; i < strlen(token); ++i) {

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

isValid = 0;

break;

}

}

if (isValid && (isdigit(token[0]) || token[0] == '-')) {

isValid = 0;

}

if (isValid) {

printf("Токен '%s' є допустимим ідентифікатором\n", token);

}

token = strtok(NULL, " ");

}

}

int main() {

char str[] = "valid \_valid\_ invalid-s valid123 \_Valid 1invalid ";

identificatorOrNot(str);

return 0;

}

*Рисунок 2: Код завдання з VNS №6*

Завдання №2: VNS Lab 8

#include <iostream>

#include <fstream>

#include <string>

#include <vector>

using namespace std;

struct Film {

string title;

string director;

string country;

string revenue;

};

void printFileContents(const string& path) {

ifstream file(path);

if (file.is\_open()) {

string line;

while (getline(file, line)) {

cout << line << endl;

}

file.close();

} else {

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

}

}

void deleteLastTwoLines(const string& path) {

ifstream inFile(path);

ofstream tempFile("temp.txt");

if (!inFile || !tempFile) {

cerr << "Error opening files." << endl;

return;

}

vector<string> lines;

string line;

while (getline(inFile, line)) {

lines.push\_back(line);

}

inFile.close();

if (lines.size() > 1) {

lines.pop\_back();

lines.pop\_back();

}

for (const string& l : lines) {

tempFile << l << endl;

}

tempFile.close();

remove(path.c\_str());

rename("temp.txt", path.c\_str());

}

void addElement(const string& path) {

Film newMovie;

cout << "Enter the title of the film: ";

getline(cin, newMovie.title);

cout << "Enter the director of the film: ";

getline(cin, newMovie.director);

cout << "Enter the country of the film: ";

getline(cin, newMovie.country);

cout << "Enter the revenue of the film: ";

getline(cin, newMovie.revenue);

ofstream outFile(path, ios::app);

if (!outFile) {

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

return;

}

outFile << newMovie.title << endl;

outFile << newMovie.director << endl;

outFile << newMovie.country << endl;

outFile << newMovie.revenue << endl;

outFile.close();

}

int main() {

string path = "t.txt";

cout << "File Contents Before Operations:\n";

printFileContents(path);

deleteLastTwoLines(path);

addElement(path);

cout << "\nFile Contents After Operations:\n";

printFileContents(path);

return 0;}

*Рисунок 3: Код завдання з VNS №8*

Завдання №3: VNS Lab 9

#include <iostream>

#include <fstream>

#include <sstream>

#include <cctype>

using namespace std;

bool vowel(char ch);

int main() {

char firstWord[255];

ofstream f1("F1.txt");

if (!f1.is\_open()) {

cout << "Failed to open file F1.txt." << endl;

return 1;

}

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

cout << "Enter " << i + 1 << " lines for F1.txt:" << endl;

string line;

getline(cin, line);

f1 << line << endl;

}

f1.close();

ifstream f1R("F1.txt");

ofstream f2("F2.txt");

if (!f1R.is\_open() || !f2.is\_open()) {

cout << "Failed to open files." << endl;

return 1;

}

f1R >> firstWord;

f1R.clear();

f1R.seekg(0, ios::beg);

string line;

int consonantCount = 0;

if (getline(f1R, line)) {

for (char ch : line) {

if (!vowel(ch) && ch != ' ') {

consonantCount++;

}

}

cout << "Number of consonants in the first line of F2: " << consonantCount << endl;

f2 << line << endl;

}

while (getline(f1R, line)) {

bool choice = true;

stringstream ss(line);

string word;

while (ss >> word) {

if (word == firstWord) {

choice = false;

break;

}

}

if (choice) {

f2 << line << endl;

}

}

f1R.close();

f2.close();

cout << "Program executed successfully. Press Enter to exit.";

cin.get();

return 0;

}

bool vowel(char ch) {

ch = tolower(ch);

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

}

*Рисунок 4: Код завдання з VNS №9*

Завдання №4: Algotester Lab 4

#include <iostream>

using namespace std;

void swap(int& a, int& b) {

int temp = a;

a = b;

b = temp;

}

int partitionUp(int arr[], int low, int high) {

int cock = arr[high];

int i = (low - 1);

for (int j = low; j <= high - 1; j++) {

if (arr[j] < cock) {

i++;

swap(arr[i], arr[j]);

}

}

swap(arr[i + 1], arr[high]);

return (i + 1);

}

void quickSortUp(int arr[], int low, int high) {

if (low < high) {

int pi = partitionUp(arr, low, high);

quickSortUp(arr, low, pi - 1);

quickSortUp(arr, pi + 1, high);

}

}

int partitionDown(int arr[], int low, int high) {

int cock = arr[high];

int i = (low - 1);

for (int j = low; j <= high - 1; j++) {

if (arr[j] > cock) {

i++;

swap(arr[i], arr[j]);

}

}

swap(arr[i + 1], arr[high]);

return (i + 1);

}

void quickSortDown(int arr[], int low, int high) {

if (low < high) {

int pi = partitionDown(arr, low, high);

quickSortDown(arr, low, pi - 1);

quickSortDown(arr, pi + 1, high);

}

}

int main() {

int minus = 0;

int n;

cin >> n;

int leftZero = 0;

int arr[10000] = {0};

int New[10000] = {0};

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

cin >> arr[i];

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

New[leftZero] = arr[i];

leftZero++;

}

}

quickSortUp(New, 0, leftZero - 1);

int leftOne=leftZero;

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

if (arr[i] % 3 == 1) {

New[leftOne] = arr[i];

leftOne++;

}

}

quickSortDown(New, leftZero, leftOne - 1);

int leftTwo=leftOne;

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

if (arr[i] % 3 == 2) {

New[leftTwo] = arr[i];

leftTwo++;

}

}

quickSortUp(New, leftOne, leftTwo - 1);

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

for(int j = 0; j<n; j++){

if(New[i]==New[j] && i != j){

New[i] = -1;

minus++;

}

}

}

cout << n-minus << endl;

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

if(New[i] != -1){

cout << New[i] << " ";

}

}

return 0;

}

Метод 2:

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

int main() {

int N;

cin >> N;

vector<int> numbers(N);

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

cin >> numbers[i];

}

vector<int> correct;

int countDivisibleBy3 = 0;

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

correct.push\_back(numbers[i]);

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

countDivisibleBy3++;

}

}

sort(correct.begin(), correct.end(), [countDivisibleBy3](int a, int b) {

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

return a > b;

}

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

});

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

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

for (int num : correct) {

cout << num << " ";

}

return 0;

}

*Рисунок 5: Код завдання з Algotester №4*

*#include <iostream>*

*#include <vector>*

*using namespace std;*

*int main() {*

*int n;*

*int q;*

*int x, y;*

*int placeOfAnalyse;*

*vector<char> arr(100, 0);*

*cin >> n;*

*int elementsToChange = 0;*

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

*cin >> arr[i];*

*}*

*cin >> q;*

*const int MAX\_ITERATIONS = 100;*

*const int REPLACEMENT\_SIZE = 9;*

*int results[MAX\_ITERATIONS];*

*vector<vector<char>> replacements(MAX\_ITERATIONS, vector<char>(REPLACEMENT\_SIZE, 0));*

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

*cin >> x >> y;*

*placeOfAnalyse = (x - 1) \* n + y - 1;*

*if (arr[placeOfAnalyse] != '0') {*

*results[i] = 1;*

*replacements[i][0] = arr[placeOfAnalyse];*

*continue;*

*}*

*vector<char> lines = {'1', '2', '3', '4', '5', '6', '7', '8', '9'};*

*for (int u = 0; u < n; u++) {*

*placeOfAnalyse = (x - 1) \* n + u;*

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

*if (arr[placeOfAnalyse] == lines[j]) {*

*lines[j] = 0;*

*}*

*}*

*}*

*for (int u = 0; u < n; u++) {*

*placeOfAnalyse = u \* n + (y - 1);*

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

*if (arr[placeOfAnalyse] == lines[j]) {*

*lines[j] = 0;*

*}*

*}*

*}*

*elementsToChange = 0;*

*for (int h = 0; h < n; h++) {*

*if (lines[h] != 0) {*

*elementsToChange++;*

*}*

*}*

*results[i] = elementsToChange;*

*int k = 0;*

*for (int j = 0; j < n && k < REPLACEMENT\_SIZE; j++) {*

*if (lines[j] != 0) {*

*replacements[i][k++] = lines[j];*

*}*

*}*

*}*

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

*cout << results[i] << endl;*

*for (int j = 0; j < results[i]; j++) {*

*cout << replacements[i][j] << " ";*

*}*

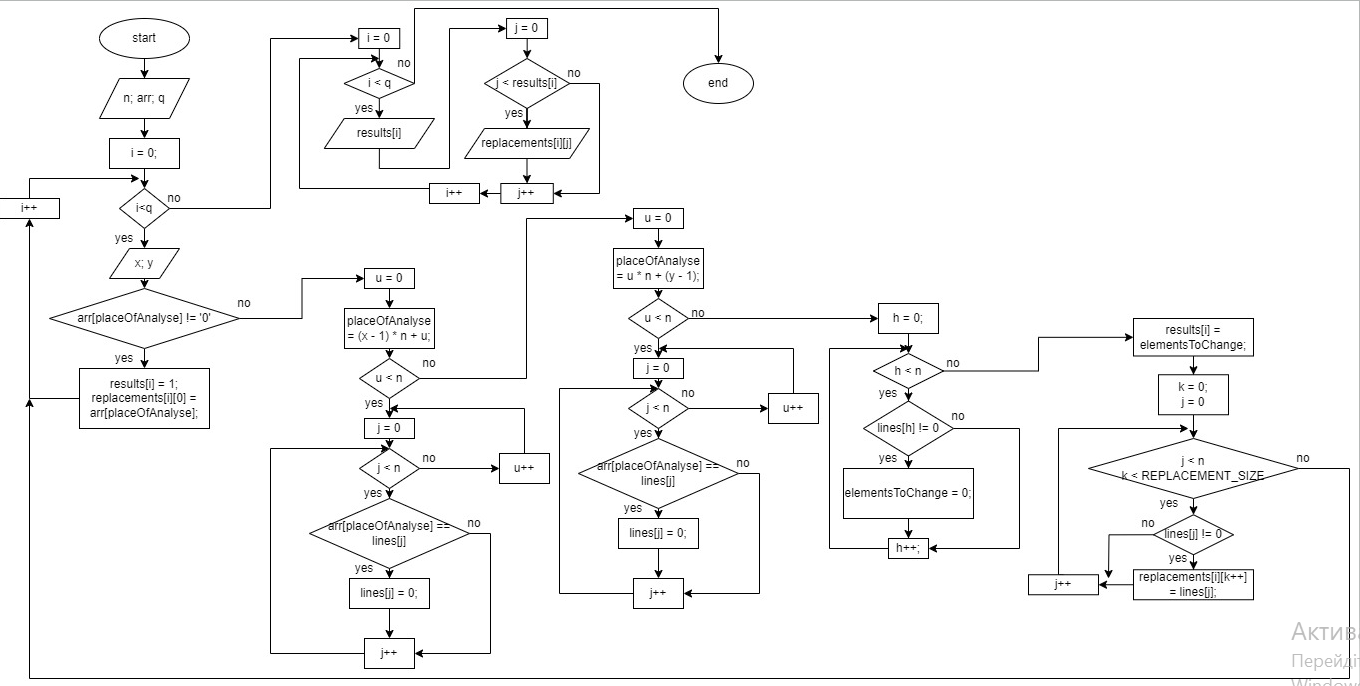
*cout << endl;*

*}*

*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 <sstream>

using namespace std;

enum FileOpResult { Success, Failure };

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

ofstream file(name);

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

return Failure;

}

file << content;

file.close();

return Success;

}

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

ifstream source(file\_from, ios::binary);

ofstream destination(file\_to, ios::binary);

if (!source.is\_open() || !destination.is\_open()) {

return Failure;

}

destination << source.rdbuf();

source.close();

destination.close();

return Success;

}

int main() {

string content;

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

getline(cin, content);

const char\* filename = "example.txt";

FileOpResult result = write\_to\_file(filename, content.c\_str());

if (result == Success) {

cout << "Content copied" << endl;

} else {

cout << "Failed to write to file." << endl;

return 1;

}

string file\_to\_copy;

cout << "Enter the name of the file to copy to: ";

getline(cin, file\_to\_copy);

result = copy\_file(filename, file\_to\_copy.c\_str());

if (result == Success) {

cout << "Content is pasted." << endl;

} else {

cout << "Failed to copy file." << endl;

return 1;

}

return 0;

}

*Рисунок 7: Код програми з практичної роботи*

Завдання №7: Self Practice Task

#include <iostream>

using namespace std;

void swap(int &a, int &b) {

int temp = a;

a = b;

b = temp;

}

void removeDuplicates(int arr[], int &n) {

int newSize = 0;

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

bool isDuplicate = false;

for (int j = 0; j < newSize; ++j) {

if (arr[i] == arr[j]) {

isDuplicate = true;

break;

}

}

if (!isDuplicate) {

arr[newSize++] = arr[i];

}

}

n = newSize;

}

void rotateArray(int arr[], int n, int k) {

k %= n;

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

int temp = arr[0];

for (int j = 0; j < n - 1; ++j) {

arr[j] = arr[j + 1];

}

arr[n - 1] = temp;

}

}

int main() {

int N, K;

cin >> N >> K;

int a[1000];

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

cin >> a[i];

}

removeDuplicates(a, N);

for (int i = 0; i < N - 1; ++i) {

for (int j = 0; j < N - i - 1; ++j) {

if (a[j] > a[j + 1]) {

swap(a[j], a[j + 1]);

}

}

}

rotateArray(a, N, K);

cout << N << endl;

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

cout << a[i] << " ";

}

return 0;

}

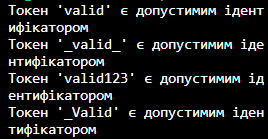
*Рисунок 8: Код програми для самостійного опрацювання*

Посилання на програми в GitHub: <https://github.com/Zimnov1/Repo/tree/main>

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

Завдання №1: VNS Lab 6

Результати виконання програми:

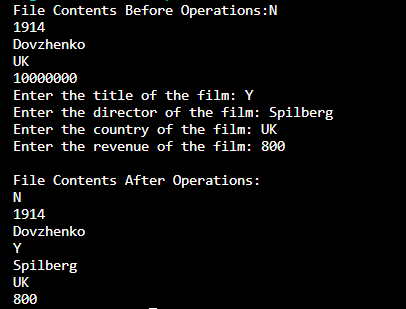


*Рисунок 9: Результати виконання програми з VNS №6*

Час затрачений на виконання завдання: 1 година

Завдання №2: VNS Lab 8

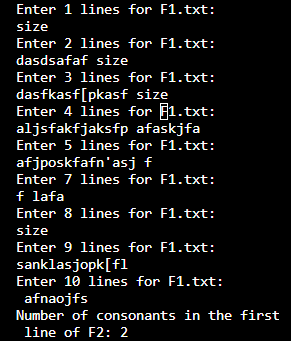
Результати виконання програми:

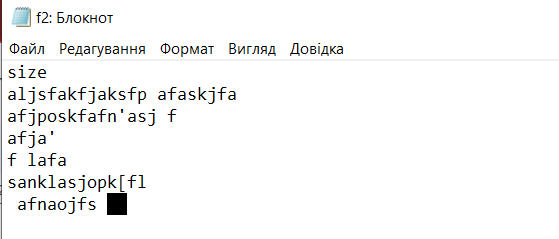


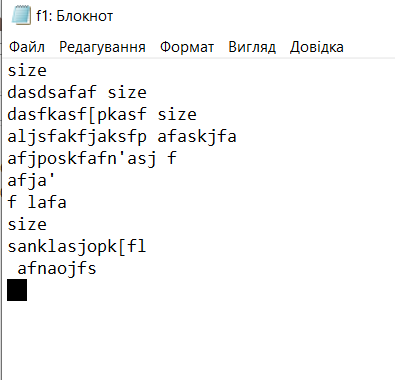
*Рисунок 10: Результати виконання програми з VNS №8*

Час затрачений на виконання завдання: 3 години

Завдання №3: VNS Lab 9



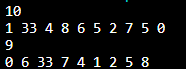




*Рисунок 11: Результати виконання програми з VNS №9*

Завдання №4: Algotester Lab 4

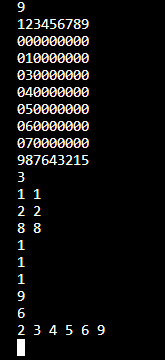
Результати виконання програми:



*Рисунок 12: Результати виконання програми з Algotester №4*

Завдання №5: Algotester Lab 6

Результати виконання програми:

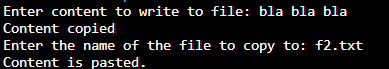


*Рисунок 13: Результати виконання програми з Algotester №6*

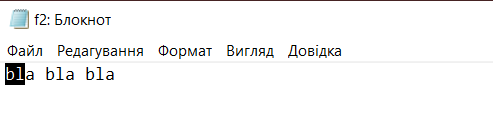
Час затрачений на виконання завдання: 5 годин

Завдання №6: Class Practice Task

Результати виконання програми:



*Рисунок 14: Результати виконання практичної роботи*

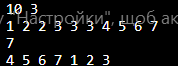


*Рисунок 15: Результати виконання практичної роботи в текстовому документі*

Час затрачений на виконання завдання: 30 хвилин

Завдання №7: Self Practice Task

Результати виконання програми:



*Рисунок 16: Результати виконання роботи для самостійного опрацювання*

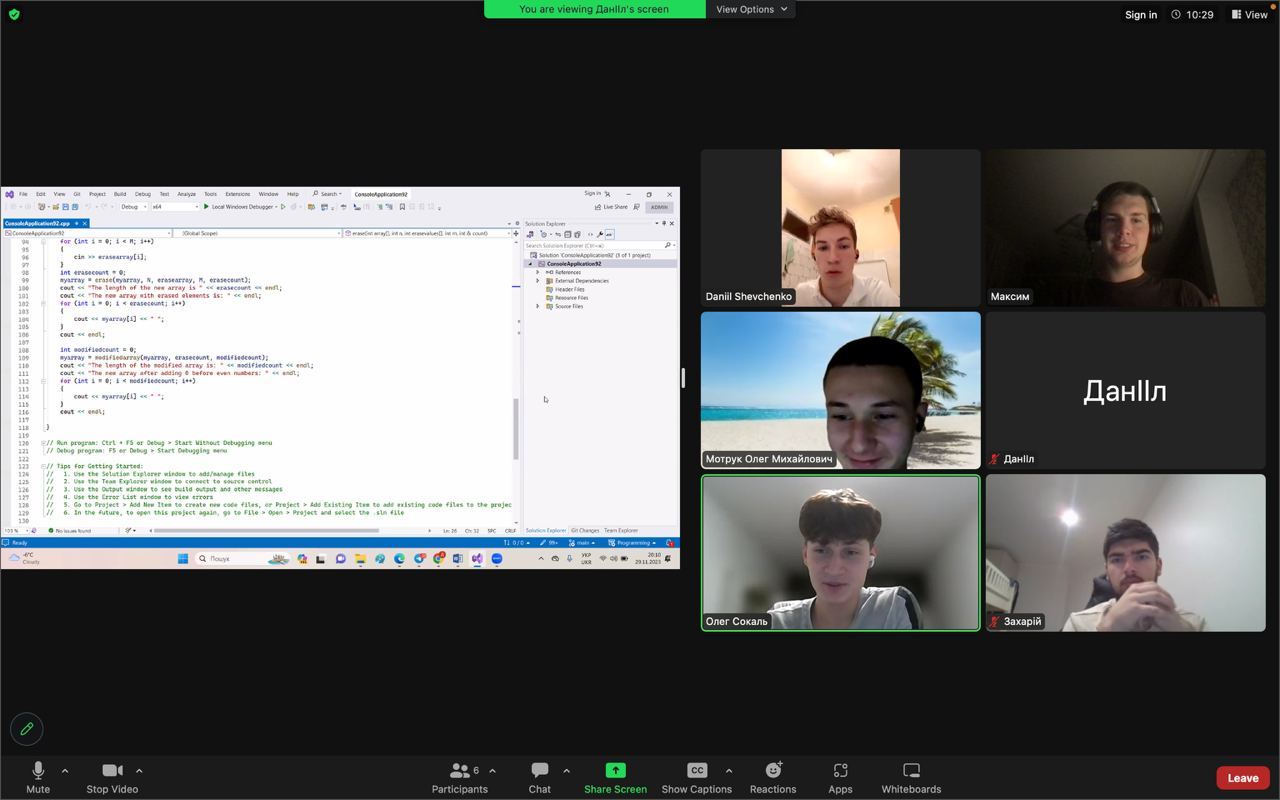
Деталі по виконанню:

* 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.Вивчив принцип роботи Quicksort.

Були онлайн зустрічі з командою, в яких ми владнали всі непорозуміння.