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

Кроляк Роман Романович

# **Тема роботи:**

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

# **Мета роботи:**

# Навчитися працювати з файлами та бібліотеками

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Робота з файлами
* Тема №2: Системи числення
* Тема №3: Стандартна бібліотека

1. Індивідуальний план опрацювання теорії:

* Тема №1: Робота з файлами
  + Джерела Інформації
    - <https://www.w3schools.com/cpp/cpp_files.asp>
    - <https://medium.com/@zekumorudoragonhatto/how-to-work-with-binary-files-in-c-520a852ee04a>
  + Що опрацьовано:
    - Створення, знищення, переіменування файлів, запис та зчитування інформації
  + Статус: Ознайомлений
  + Початок опрацювання теми: 12.12.2023
  + Звершення опрацювання теми: 13.12.2023
* Тема №2: Системи числення
  + Джерела Інформації:
    - Пара з архітектури комп’ютера
  + Що опрацьовано:
    - Десяткову, двійкову, вісімкову та шіснадцяткову системи числення
  + Статус: Ознайомлений
  + Початок опрацювання теми: 06.11.2023
  + Звершення опрацювання теми: 06.11.2023
* Тема №3: Стандартна бібліотека
  + Джерела Інформації
    - <https://www.geeksforgeeks.org/sort-c-stl/t>
    - <https://www.geeksforgeeks.org/stdpartition-in-c-stl/>
    - <https://www.geeksforgeeks.org/stdunique-in-cpp/>
  + Що опрацьовано:
    - Алгоритми стандартної бібліотеки
  + Статус: Ознайомлений
  + Початок опрацювання теми: 03.12.2023
  + Звершення опрацювання теми: 13.12.2023

# **Виконання роботи:**

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

Завдання №1 VNS Lab 6

* Варіант завдання: 2
* 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 не містять голосних букв.

Завдання №2 VNS Lab 8

* Варіант завдання: 2
* 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 "Співробітник":
  + прізвище, ім'я, по батькові;
  + посада;
  + рік народження;
  + заробітна плата.
* Знищити елемент із зазначеним прізвищем, додати елемент після елемента
* із зазначеним номером.

Завдання №3 VNS Lab 9

* Варіант завдання: 2

# Створити текстовий файл F1 не менше, ніж з 10 рядків і записати в нього інформацію Виконати завдання.

# 1) Скопіювати у файл F2 тільки ті рядки з F1, які починаються з букви «А».

# 2) Підрахувати кількість слів в F2.

Завдання №4 Algotester Lab 4

* Варіант завдання: 3
* Вам дано масив, який складається з N додатніх цілих чисел.

Ваше завдання - розділити його на три частини, по остачі від ділення на 3, по зростанню остачі (тобто спочатку йдуть числа, у яких остача 0, далі числа з остачею 1 і тоді нарешті числа з остачею 2).

Далі необхідно ті елементи, остача від ділення на 3 яких парна посортувати по зростанню, а ті, у яких остача 1 - по спаданню.

Після цього видаліть усі дублікати з масиву.

Виведіть результуючий масив.

Завдання №5 Algotester Lab 6

* Варіант завдання: 3
* У Клінта в черговий раз виключилось світло і йому немає чим зайнятися. Так як навіть це не заставить його подивитися збережені відео про програмування на ютубі - він вирішив придумати свою гру на основі судоку.

Гра виглядає так:

Є поле розміром N × N, в якому частина клітинок заповнена цифрами, а частина клітинок пусті (позначаються нулем). Також у нього є Q пар координат X та Y.

Завданням гри є написати до кожної координати скільки чисел туди можна вписати (якщо вона пуста) і які це числа (обов’язково в посортовані по зростанню!). В клітинку можна вписати лише ті числа, які не зустрічаються в рядку та стовбці, які перетинаються у цій клітинці.

Під час гри поле не міняється!

Також необовязково, щоб це було валідне судоку! Якщо є клітинка, в яку не можна вписати ніяку цифру - виведіть 0.

Також допускаються рядки та стовпці, в яких цифра записана кілька разів.

Завдання №6 Class Practice Task

* 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.
* 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.

Завдання №7 Self Practice Task

* <https://algotester.com/uk/ContestProblem/DisplayWithEditor/134777>
* Петрик любить грати в ігри, де світ складається з кубів. Сьогодні він вирішив написати свій варіант такої гри.

Петрик хоче, щоб у цій грі можна було програмувати нескладні механізми на основі електричного сигналу. Вважається, що сигнал має певну додатну силу, яка позначається цілим числом. Також, сигнал може бути відсутнім, що позначається як сигнал силою 0. Петрик додав в гру три блоки, які по-різному обробляють електричний сигнал:

Кабель отримує сигнал силою k і видає сигнал силою max(k−1,0).

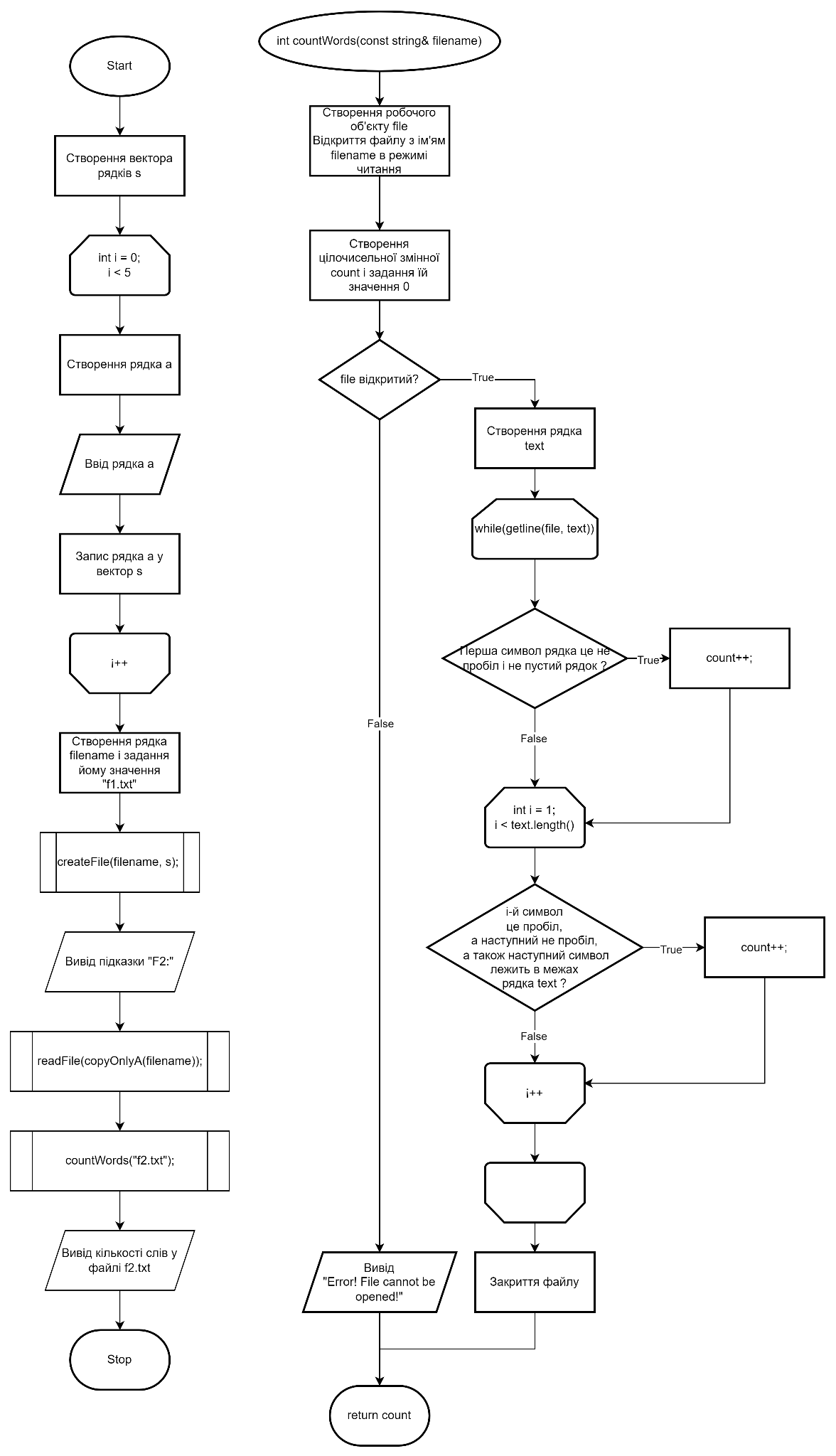
Повторювач підсилює сигнал до рівня 15 при ненульовому вхідному сигналі, і не видає сигнал при його відсутності. Формально, нехай сила вхідного сигналу k. Тоді повторювач видає сигнал силою 15, якщо k≥1, і сигнал силою 0, якщо k=0.

Змінювач не видає сигнал при ненульовому вхідному сигналі та видає сигнал силою 15 при відсутності вхідного сигналу. Формально, нехай сила вхідного сигналу k. Тоді змінювач видає сигнал силою 0, якщо k≥1, і сигнал силою 15, якщо k=0.

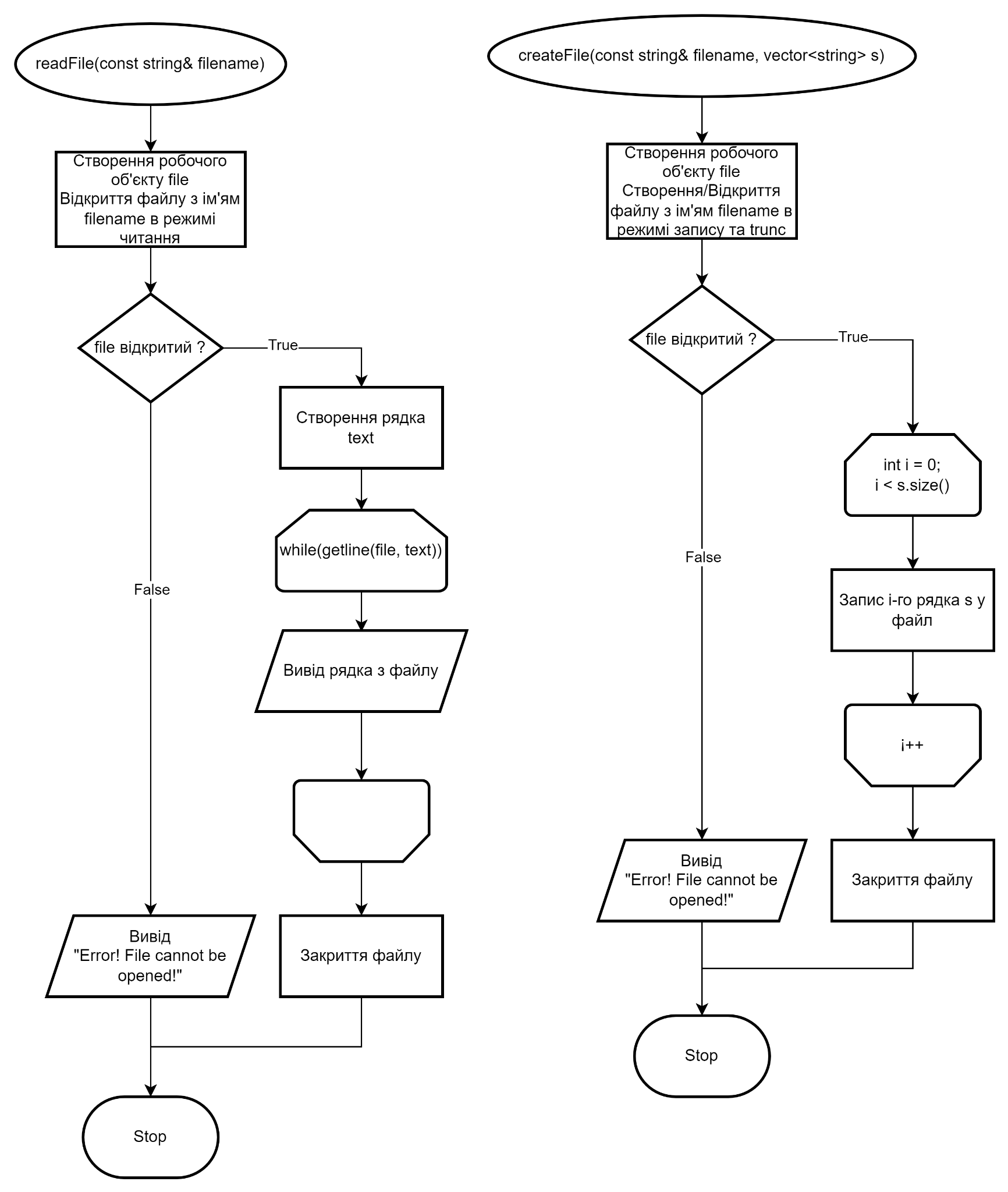
Петрик задає вам електричну лінію, в якій кабелі, повторювачі і змінювачі позначені літерами k, p, z відповідно. На перший елемент лінії подано сигнал силою 15. Допоможіть Петрику знайти вихідний сигнал останнього блоку в лінії.

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

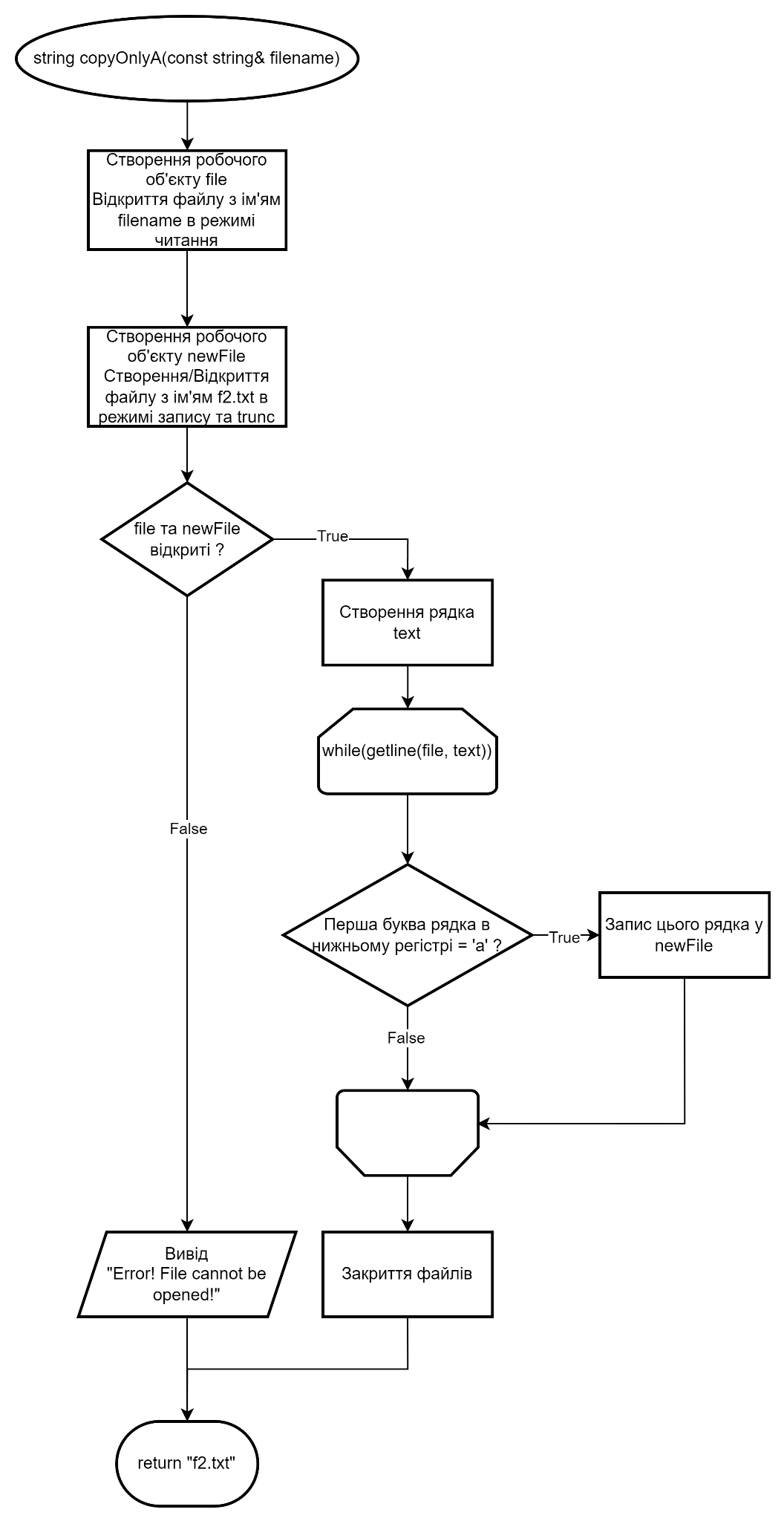
Програма №3 VNS Lab 9

* 

*Рисунок 1 Блок схема для програми №3*

* 

*Рисунок 2 Блок схема для програми №3*

* 

*Рисунок 3 Блок схема для програми №3*

* Планований час на реалізацію: 80 хв

## **3. Конфігурація середовища до виконання завдань:**

## Змін у конфігурацію середовища не вносив.

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

Завдання №1 VNS Lab 6

#include <iostream>

#include <string>

using namespace std;

int main() {

char s[255];

gets(s);

bool out = true;

string word;

int size = 0;

for (char i : s) size++;

for (char i : s) {

word.push\_back(i);

switch (i) {

case 'a':case 'e':case 'i':case 'o':case 'u':

out = false;

break;

case ' ':

if (out) cout << word;

word = "";

out = true;

break;

default:

if (i == s[size-1] && out) {

cout << word;

out = false;

}

break;

}

}

return 0;

}

https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/898/files#diff-bdb70defcd449413136951578b0e32d2b9168c6af7b47e3f6258e104edad3401

Завдання №2 VNS Lab 8

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

struct employee {

string name;

string middleName;

string lastname;

string position;

int bYear{};

int salary{};

};

void printEmployee(const employee& a){

cout << "Name: " << a.name << endl;

cout << "Middle name: " << a.middleName << endl;

cout << "Lastname: " << a.lastname << endl;

cout << "Position: " << a.position << endl;

cout << "Birth year: " << a.bYear << endl;

cout << "Salary: " << a.salary << "$\n\n";

}

void createFile (const string& filename, employee cos[], int n){

ofstream file(filename, ios::binary | ios:: trunc);

if (file.is\_open()) {

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

file.write((char\*)& cos[i], sizeof(employee));

}

file.close();

}

else cout << "Error! File cannot be opened!" << endl;

}

void readFile (const string& filename){

ifstream file(filename, ios::binary);

if (file.is\_open()) {

employee temp;

while(file.read((char\*)& temp, sizeof(employee))){

printEmployee(temp);

}

file.close();

}

else cout << "Error! File cannot be opened!" << endl;

}

void fireEmployee (const string& filename, const string& lastname){

ofstream newFile("temp.bin", ios::binary | ios:: trunc);

ifstream file(filename, ios::binary);

if (file.is\_open()) {

employee temp;

while(file.read((char\*)& temp, sizeof(employee))){

if(temp.lastname != lastname) newFile.write((char\*)& temp, sizeof(employee));

}

file.close();

newFile.close();

remove(filename.c\_str());

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

}

else cout << "Error! File cannot be opened!" << endl;

}

void hireEmployeeAt (const string& filename, int n, const employee& a){

ofstream newFile("temp.bin", ios::binary | ios:: trunc);

ifstream file(filename, ios::binary);

if (file.is\_open()) {

employee temp;

int i = 0;

while (file.read((char\*)&temp, sizeof(employee))) {

newFile.write((char\*)&temp, sizeof(employee));

if (i == n) {

newFile.write((char\*)&a, sizeof(employee));

}

i++;

}

file.close();

newFile.close();

remove(filename.c\_str());

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

}

else cout << "Error! File cannot be opened!" << endl;

}

int main() {

employee employees[] = {

{"Dallas", "Marion", "Patterson", "Data scientist", 1999, 3000},

{"Pavlina", "Puck", "Stewart", "C++ Developer", 2001, 4000},

{"Nicholas", "Riley", "West", "Project manager", 2000, 2000}

};

int employeesNum = sizeof(employees)/sizeof(employee);

string filename = "employees.bin";

createFile(filename, employees, employeesNum);

cout << "The initial content of the file:\n";

readFile(filename);

fireEmployee(filename, "Patterson");

employee newEmp = {"Sebastian", "Halifax", "Murphy", "Data analyst", 1998, 2700};

hireEmployeeAt(filename, 0, newEmp);

cout << "The modified content of the file:\n";

readFile(filename);

return 0;

}

https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/898/files#diff-435cf0cd6eaaa15c4b4aabf3209fa1ee00c4198993377ad3d39f36d7509971ff

Завдання №3 VNS Lab 9

#include <iostream>

#include <fstream>

#include <string>

#include <vector>

using namespace std;

void createFile (const string& filename, vector<string> s){

ofstream file(filename, ios:: trunc);

if (file.is\_open()) {

for (int i = 0; i < s.size(); i++)

{

file << s[i] << endl;

}

file.close();

}

else cout << "Error! File cannot be opened!" << endl;

}

void readFile (const string& filename){

ifstream file(filename);

if (file.is\_open()) {

string text;

while (getline(file, text)) {

cout << text << endl;

}

file.close();

}

else cout << "Error! File cannot be opened!" << endl;

}

string copyOnlyA (const string& filename){

ifstream file(filename);

ofstream newFile("f2.txt", ios:: trunc);

if (file.is\_open() && newFile.is\_open()) {

string text;

while (getline(file, text)) {

if (tolower(text[0]) == 'a') newFile << text << endl;

}

file.close();

newFile.close();

}

else cout << "Error! File cannot be opened!" << endl;

return "f2.txt";

}

int countWords (const string& filename){

ifstream file(filename);

int count = 0;

if (file.is\_open()) {

string text;

while (getline(file, text)) {

if (text[0] != \*" " && text[0] != \*"") count++;

for (int i = 1; i < text.length(); i++)

if (text[i] == \*" " && text[i+1] != \*" " && i+1 < text.length()) count++;

}

file.close();

}

else cout << "Error! File cannot be opened!" << endl;

return count;

}

int main(){

vector<string> s;

for (int i = 0; i < 5; i++){

string a;

getline(cin, a);

s.push\_back(a);

}

string filename = "f1.txt";

createFile(filename, s);

cout << "\nF2:\n";

readFile(copyOnlyA(filename));

cout << "\nWord count F2: " << countWords("f2.txt");

return 0;

}

https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/898/files#diff-71a6176381fc57a83ee1007388bf149eb18655e73538436b1389196fc88920c2

Завдання №4.1 Algotester Lab 4

#include <iostream>

#include <algorithm>

using namespace std;

int main(){

int n;

cin >> n;

int a[n];

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

cin >> a[i];

auto partitionPointZero = partition(a, a + n, [](int num) { return num % 3 == 0; });

auto partitionPointOne = partition(partitionPointZero, a + n, [](int num) { return num % 3 == 1; });

sort(a, partitionPointZero);

sort(partitionPointZero, partitionPointOne, greater<int>());

sort(partitionPointOne, a + n);

auto newEnd = unique(a, a + n);

int newSize = distance(a, newEnd);

cout << newSize << endl;

for (auto i = a; i != newEnd; ++i) {

cout << \*i << " ";

}

return 0;

}

https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/898/files#diff-c007bcfc8ca36a5c0e5e59f3b91622fa9540dce491f4fde0cfb8e4e4263f847d

Завдання №4.2 Algotester Lab 4 variant 2

#include <iostream>

#include <vector>

using namespace std;

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

int temp = a;

a = b;

b = temp;

}

int partition(vector<int>& arr, int low, int high, bool ascending = true) {

int pivot = arr[high];

int i = low - 1;

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

if (arr[j] <= pivot && ascending) {

++i;

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

}

else if(arr[j] >= pivot && !ascending){

++i;

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

}

}

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

return i + 1;

}

void quickSort(vector<int>& arr, int low, int high, bool ascending = true) {

if (low < high) {

int pivotIndex = partition(arr, low, high, ascending);

quickSort(arr, low, pivotIndex - 1, ascending);

quickSort(arr, pivotIndex + 1, high, ascending);

}

}

void divideArray(int originalArray[], int arraySize, vector<int>& subarray1, vector<int>& subarray2, vector<int>& subarray3) {

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

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

subarray1.push\_back(originalArray[i]);

} else if (originalArray[i] % 3 == 1) {

subarray2.push\_back(originalArray[i]);

} else {

subarray3.push\_back(originalArray[i]);

}

}

}

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

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

for (int j = i + 1; j < size;) {

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

for (int k = j; k < size - 1; ++k) {

arr[k] = arr[k + 1];

}

--size;

} else {

++j;

}

}

}

}

int main() {

int n;

cin >> n;

int a[n];

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

cin >> a[i];

vector<int> sub1;

vector<int> sub2;

vector<int> sub3;

divideArray(a, n, sub1, sub2, sub3);

quickSort(sub1, 0, sub1.size()-1);

quickSort(sub2, 0, sub2.size()-1, false);

quickSort(sub3, 0, sub3.size()-1);

vector<int> res;

res.insert(res.end(), sub1.begin(), sub1.end());

res.insert(res.end(), sub2.begin(), sub2.end());

res.insert(res.end(), sub3.begin(), sub3.end());

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

a[i] = res[i];

removeDuplicates(a, n);

cout << n << endl;

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

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

return 0;

}

https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/898/files#diff-7fbf00a91986dd0983add7d3dc7bc99e2e20b2918b223478bc5054a2542334c2

Завдання №5 Algotester Lab 6

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

void answer(int x, int y, int a[][10], int n, vector<int> avaible\_nums, vector<int>& answers){

if (a[x-1][y-1] != 0)

{

answers.push\_back(a[x-1][y-1]);

}

else{

for (int i : avaible\_nums)

{

bool good = true;

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

{

if (i == a[x-1][j])

{

good = false;

}

}

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

{

if (i == a[j][y-1])

{

good = false;

}

}

if (good) answers.push\_back(i);

}

}

}

int main(){

int n, q;

cin >> n;

int a[10][10];

char b[10][10];

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

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

cin >> b[i][j];

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

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

a[i][j] = int(b[i][j] - 48);

cin >> q;

int cors [q][2];

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

cin >> cors[i][0] >> cors[i][1];

vector<int> avaible\_nums;

for (int i = 1; i <= n; i++)

avaible\_nums.push\_back(i);

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

{

vector<int> answers;

answer(cors[i][0],cors[i][1],a,n,avaible\_nums,answers);

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

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

for (int i : answers)

cout << i << " ";

cout << "\n\n";

}

return 0;

}

https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/898/files#diff-bb5cb9654413c1996b32552a6c9dcba1d8103f633dbed178a9401b0f69fbbf52

Завдання №6 Class Practice Task

#include <iostream>

#include <fstream>

using namespace std;

enum FileOpResult { Success, Failure};

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

if (name == nullptr) return Failure;

string name\_s;

while(\*name != '\0'){

name\_s.push\_back(\*name);

++name;

}

ofstream file(name\_s, ios::trunc);

if (file.is\_open()) {

if (content == nullptr) return Failure;

while (\*content != '\0') {

file << \*content;

++content;

}

file.close();

return Success;

}

else return Failure;

}

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

if (file\_from == nullptr || file\_to == nullptr) return Failure;

string file\_from\_s;

while(\*file\_from != '\0'){

file\_from\_s.push\_back(\*file\_from);

++file\_from;

}

string file\_to\_s;

while(\*file\_to != '\0'){

file\_to\_s.push\_back(\*file\_to);

++file\_to;

}

ifstream file\_from\_f(file\_from\_s);

ofstream file\_to\_f(file\_to\_s, ios::trunc);

if (file\_from\_f.is\_open() && file\_to\_f.is\_open()) {

string text;

while (getline(file\_from\_f, text)) {

file\_to\_f << text << endl;

}

file\_from\_f.close();

file\_to\_f.close();

return Success;

}

else return Failure;

}

int main(){

char name[50], content[50];

gets\_s(name, 50);

gets\_s(content, 50);

(write\_to\_file(name, content)) ? cout << "Failure\n" : cout << "Success\n";

char file\_from[50], file\_to[50];

gets\_s(file\_from, 50);

gets\_s(file\_to, 50);

(copy\_file(file\_from, file\_to)) ? cout << "Failure" : cout << "Success";

return 0;

}

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/898/files#diff-fd940351e62df6f3662390fa8539ea2ab3ee3db2154d2ee138edb6332e82dab2>

Завдання №7 Self Practice Task

#include <iostream>

#include <string>

using namespace std;

int main() {

int n;

string s;

cin >> n;

cin >> s;

int power = 15;

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

if (s[i] == 'k' && power != 0) power--;

else if (s[i] == 'k' && power == 0) power = 0;

else if (s[i] == 'p' && power >= 1) power = 15;

else if (s[i] == 'p' && power == 0) power = 0;

else if (s[i] == 'z' && power == 0) power = 15;

else if (s[i] == 'z' && power >= 1) power = 0;

}

cout << power;

return 0;

}

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/898/files#diff-5d2f6c2b3100e13cac7c5504f7280703a6b76e141851a575d9707fc6116c6118>

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

Завдання №1 VNS Lab 6



*Рисунок 4 Результат виконання завдання №1*

Час затрачений на виконання завдання: 30 хв

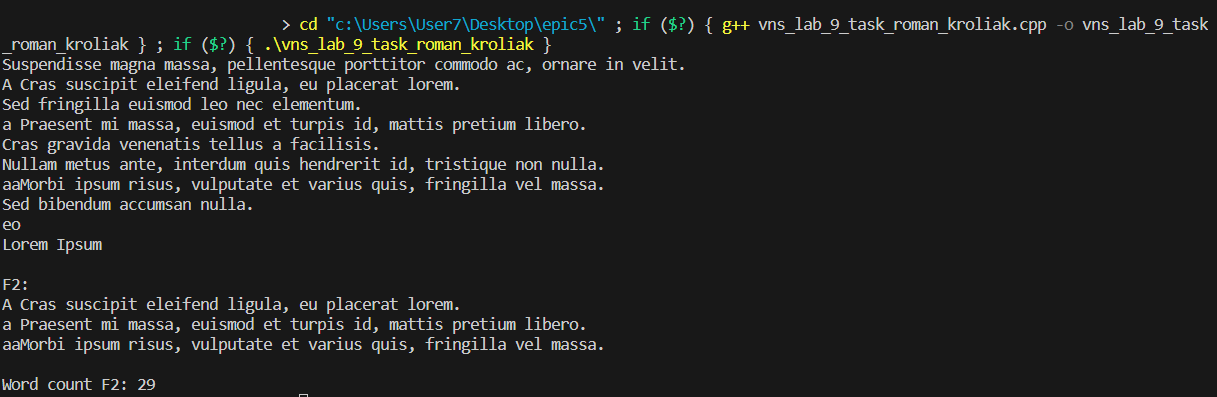
Завдання №2 VNS Lab 8

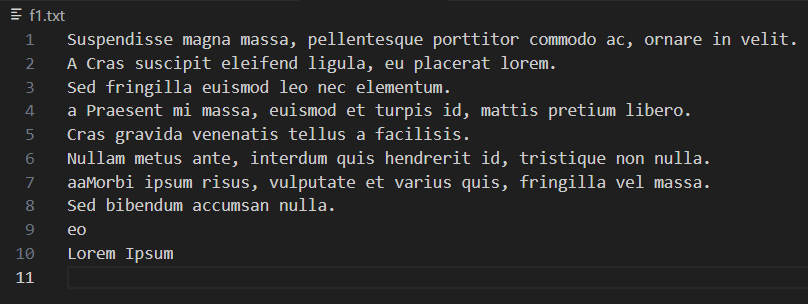


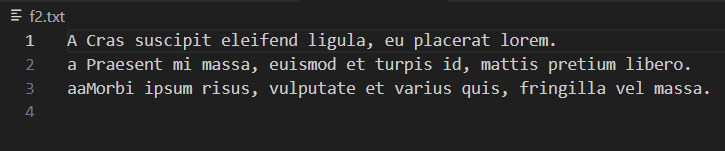
*Рисунок 5 Результат виконання завдання №2*

Час затрачений на виконання завдання: 2 год

Завдання №3 VNS Lab 9



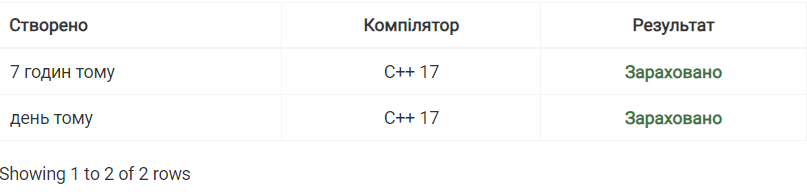
*Рисунок 6 Результат виконання завдання №3*



*Рисунок 7 Результат виконання завдання №3 Рисунок 8 Результат виконання завдання №3*

Час затрачений на виконання завдання: 30хв

Завдання №4 Algotester Lab 4



*Рисунок 9 Результат виконання завдання №4*

Час затрачений на виконання завдання: 5 год

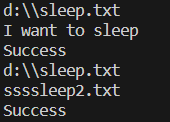
Завдання №5 Algotester Lab 6



*Рисунок 10 Результат виконання завдання №5*

Час затрачений на виконання завдання: 2 год

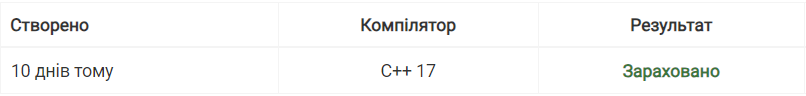
Завдання №6 Class Practice Task



*Рисунок 11 Результат виконання завдання №6*

Час затрачений на виконання завдання: 45 хв

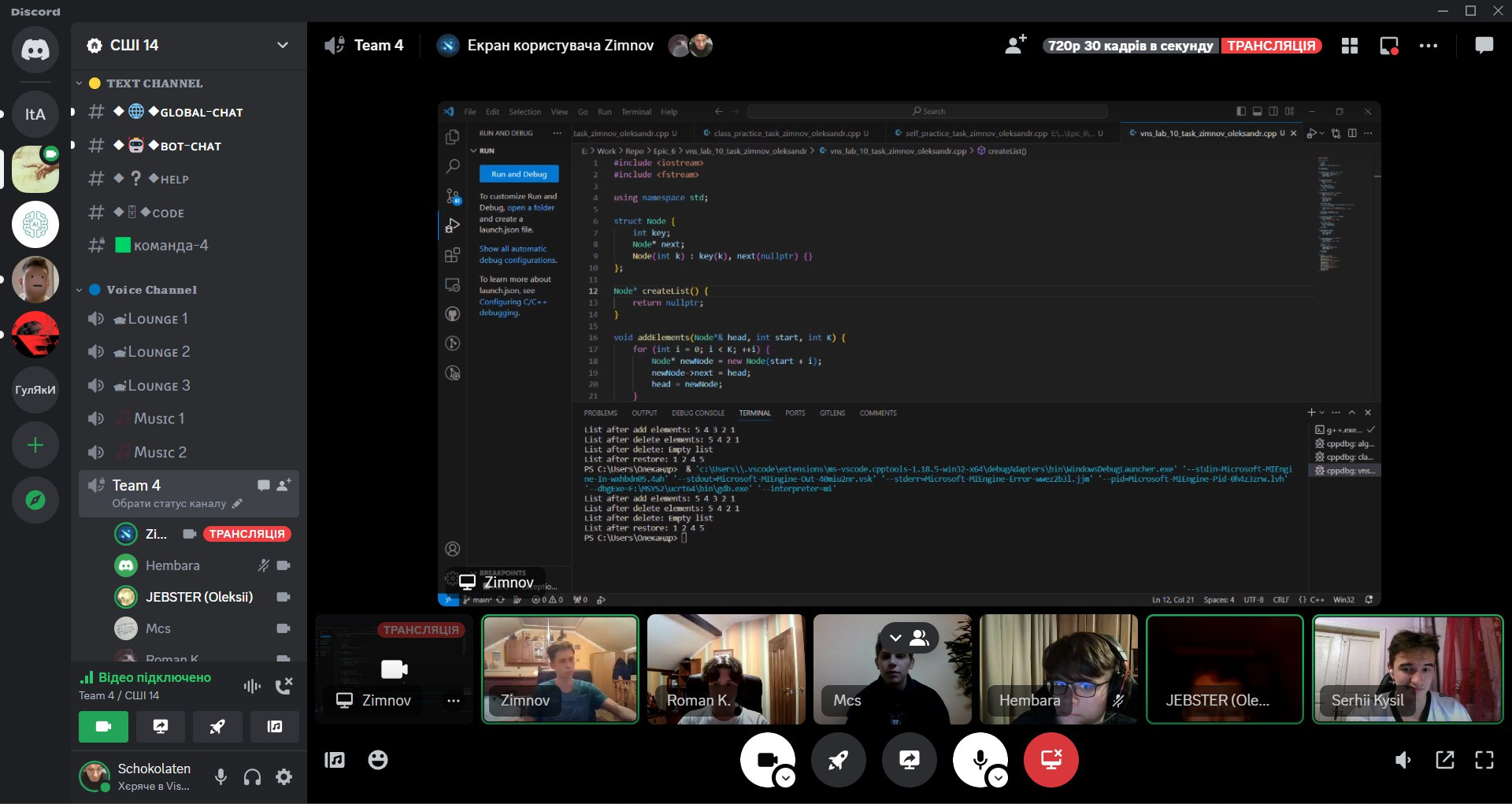
Завдання №7 Self Practice Task



*Рисунок 12 Результат виконання завдання №7*

Час затрачений на виконання завдання: 30 хв

**6. Кооперація з командою:**

* 

# **Висновки:**

Протягом п’ятого епіку я здобув досвід роботи з файлами, рядками та структурами.