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 та рядкові змінні. Текстові файли. Стандартна бібліотека. Створення й використання бібліотек.»

***Виконала:***

студент групи ШІ-12

Пасічко Софія Андріївна

# **Тема роботи:**

Ознайомлення з роботою з файлами в С++. Принцип роботи з бінарними та текстовими файлами.

# **Мета роботи:**

Теоретично та практично освоїти роботу з файлами.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Робота з файлами.
* Тема №2: Бінарні й текстові файли.
* Тема №3: Робота з символьними та рядковими змінними.

1. Індивідуальний план опрацювання теорії:

* Тема №1: Робота з файлами.
  + Джерела Інформації
  + <https://youtu.be/FeNqHytI0fA?si=mErTTwb3H348xzXO>
  + Що опрацьовано:
    - Створення, видалення, перейменування файлів, внесення, зчитування інформації та ін.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 30.11
  + Звершення опрацювання теми: 01.12
* Тема №2: Бінарні й текстові файли.
  + Джерела Інформації:
  + <https://www.easeus.com/knowledge-center/what-is-the-difference-between-binary-and-text-files.html>
  + Що опрацьовано:
    - Різниця між текстовими і бінарними файлами.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 30.11
  + Звершення опрацювання теми: 01.12
* Тема №3 Робота з символьними та рядковими змінними.
  + Джерела Інформації:
  + <http://www.kievoit.ippo.kubg.edu.ua/kievoit/2016/73_C++/index.html>
  + Що опрацьовано:
    - Основні методи роботи з символьними та рядковими змінними.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 30.11
  + Звершення опрацювання теми: 03.12

# **Виконання роботи:**

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

Завдання №1 VNS Lab 6.

Варіант 13.

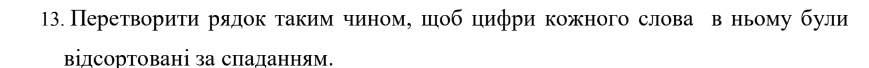
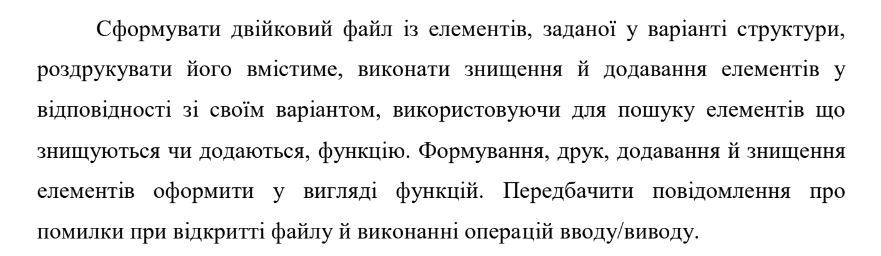


Рисунок 1. Умова завдання 1.

Завдання №2 VNS Lab 8.

Варіант 13.



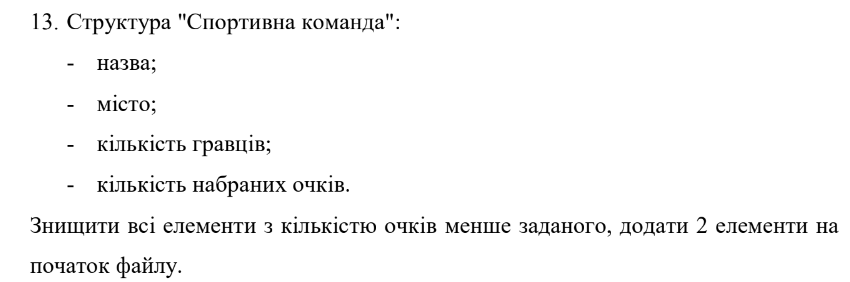


Рисунок 2. Умова завдання 2.

Завдання №3 VNS Lab 9.

Варіант 3.

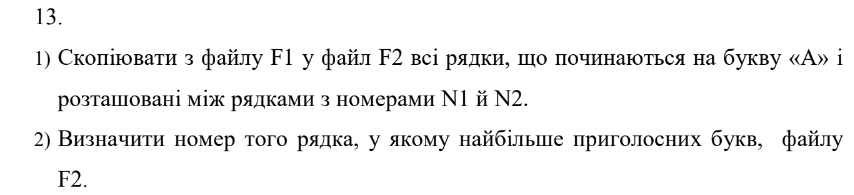


Рисунок 3. Умова завдання 3.

Завдання №4 Algotester Lab 4.

Варіант 3.

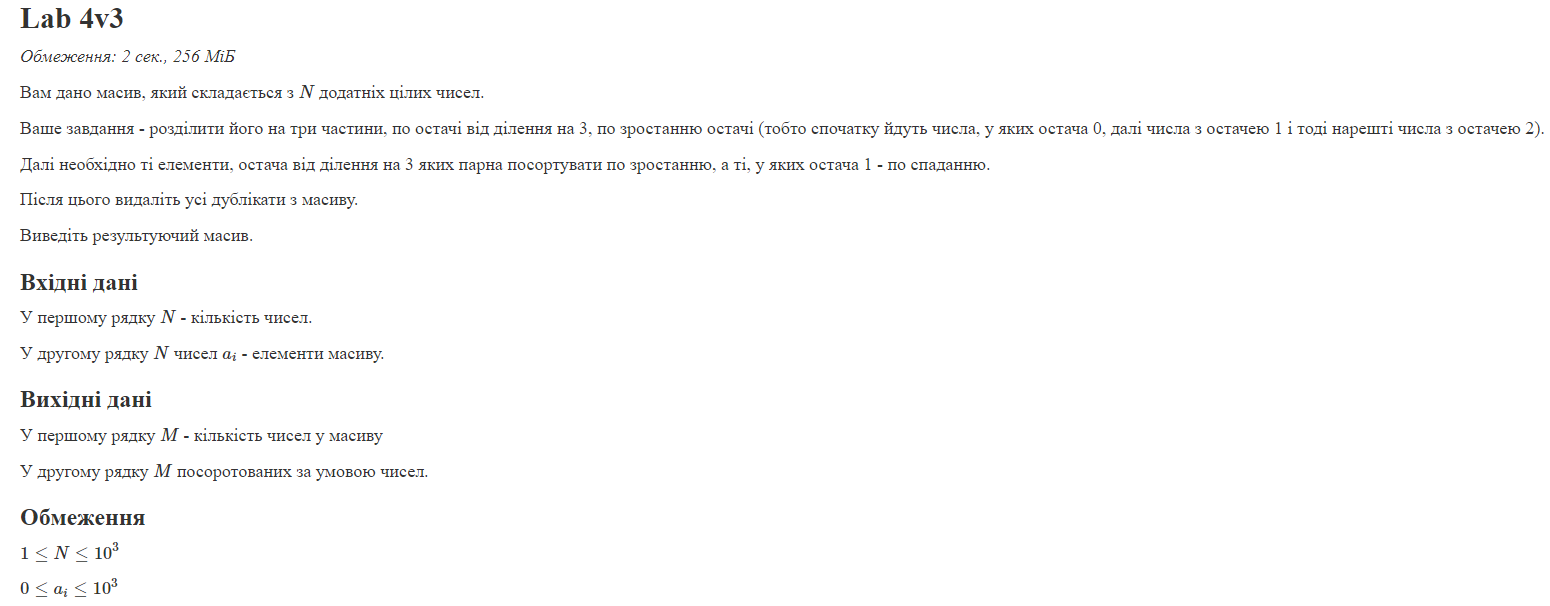


Рисунок 4. Умова завдання 4.

Важливо врахувати обмеження під час імплементації програми.

Завдання №5 Algotester Lab 6.

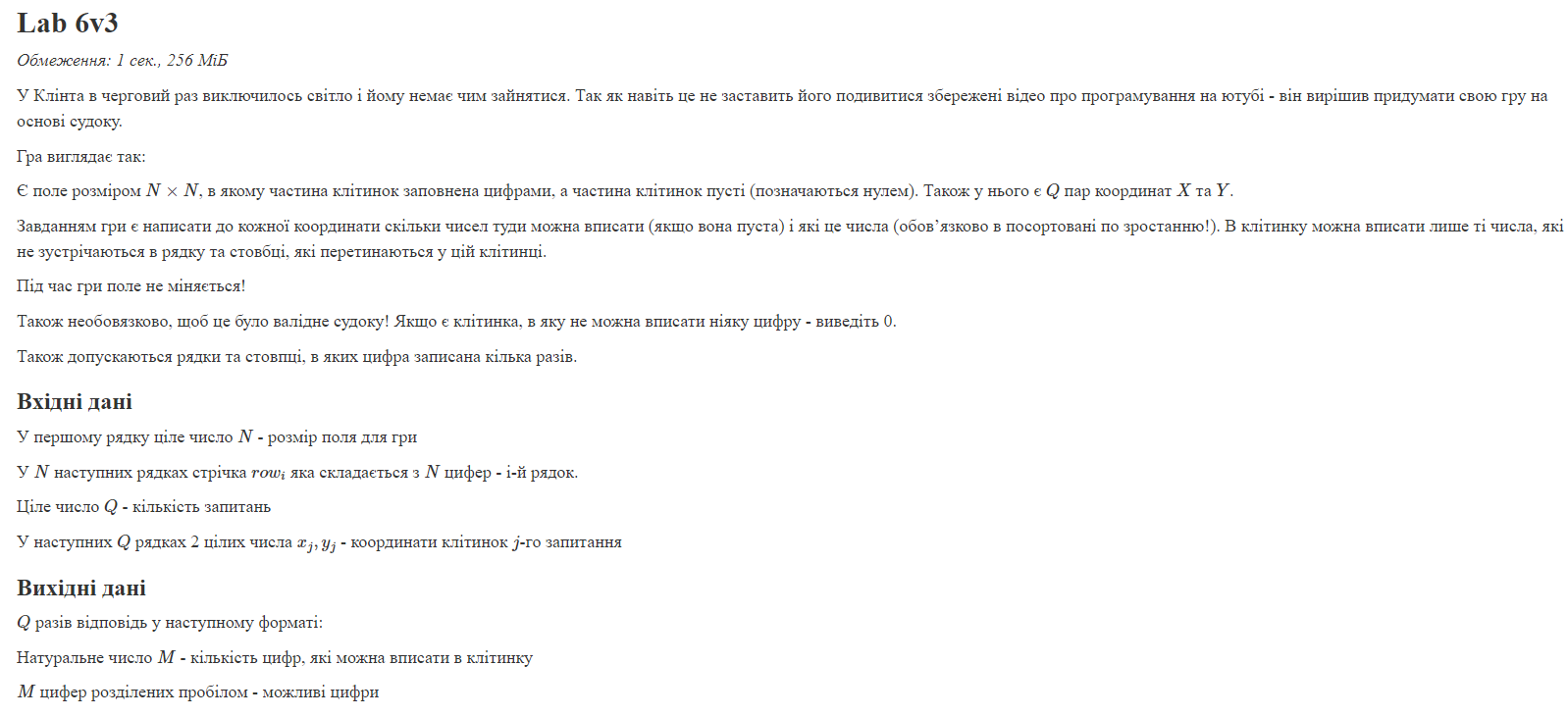


Рисунок 5. Умова завдання 5.

Важливо врахувати обмеження під час імплементації програми.

Завдання №6 Class Practice Task 1.

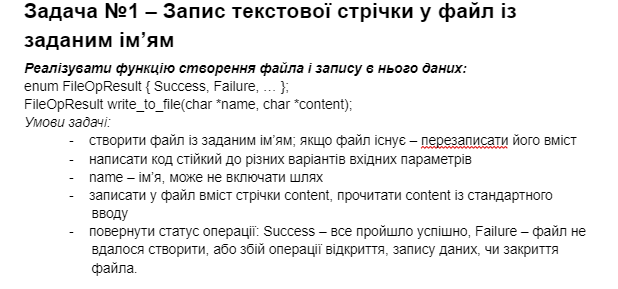
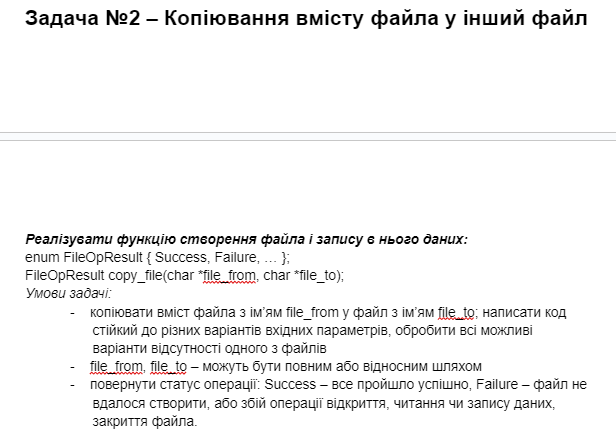


Рисунок 6. Умова завдання 6.

Завдання №7 Class Practice Task 2.



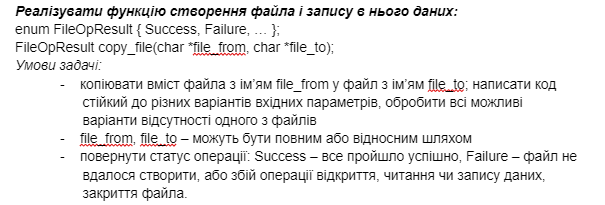


Рисунок 7. Умова завдання 7.

Завдання №8 Self-Practice Task.

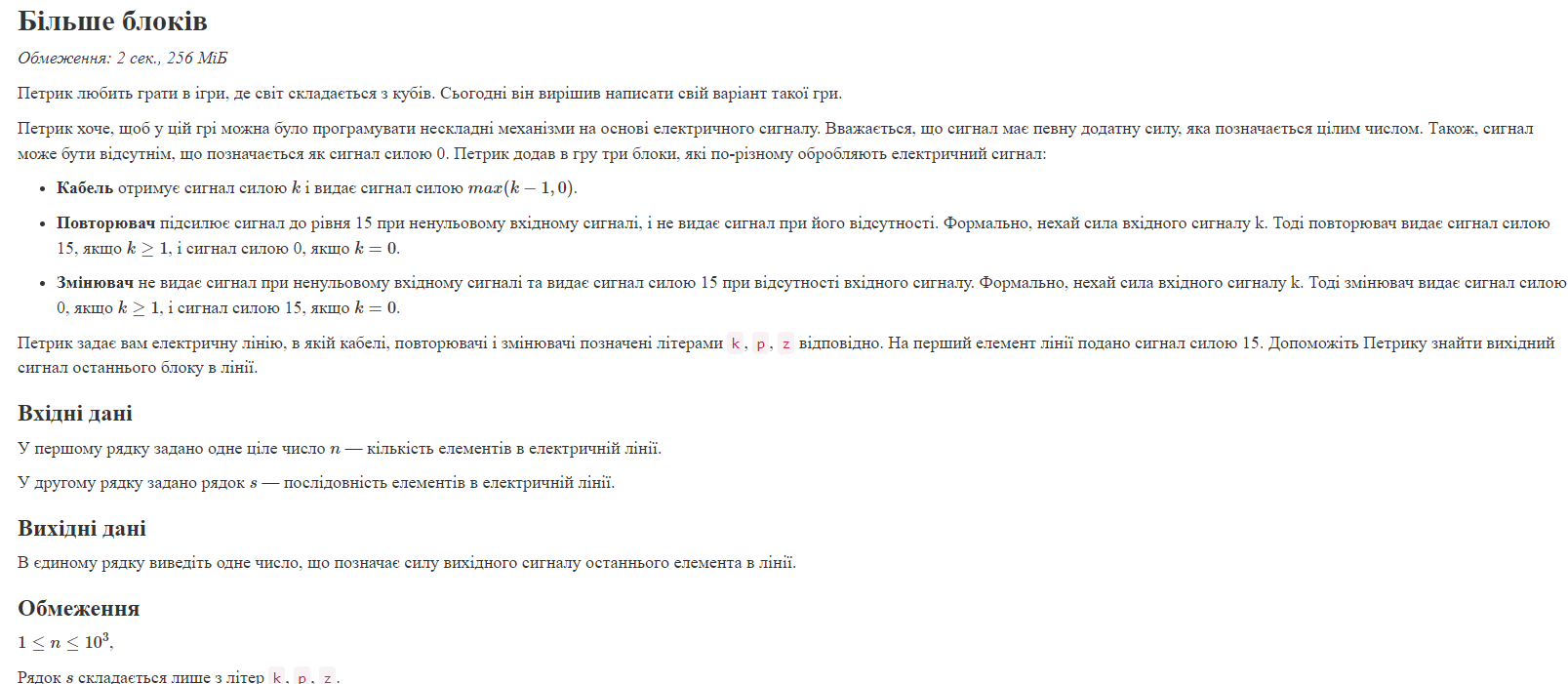


Рисунок 8. Умова завдання 8.

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

Програма №1 VNS Lab 6.

Планований час на виконання – 1,5 год

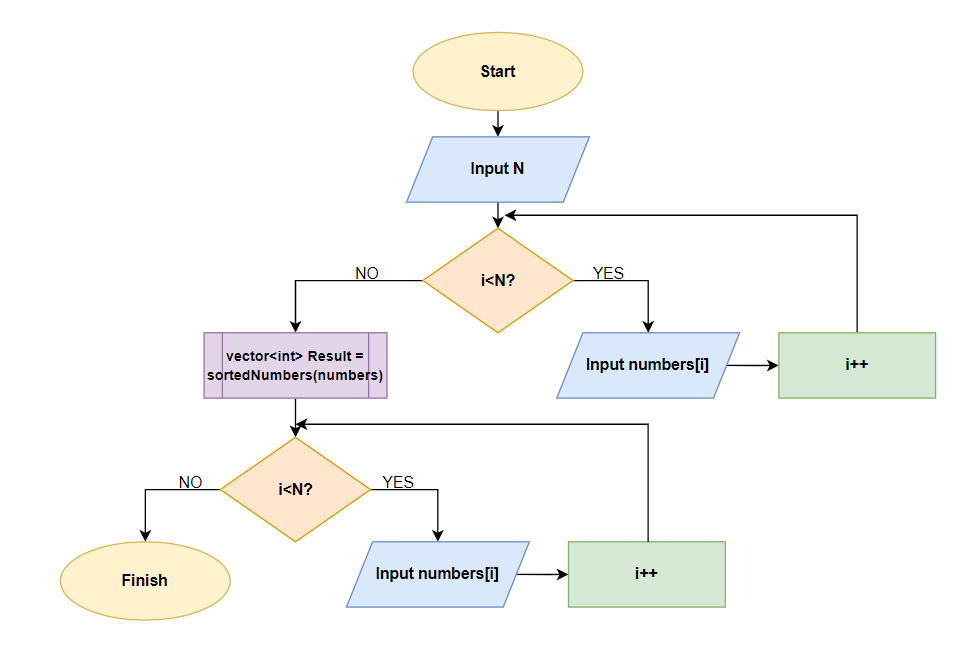
Програма №2 VNS Lab 8.

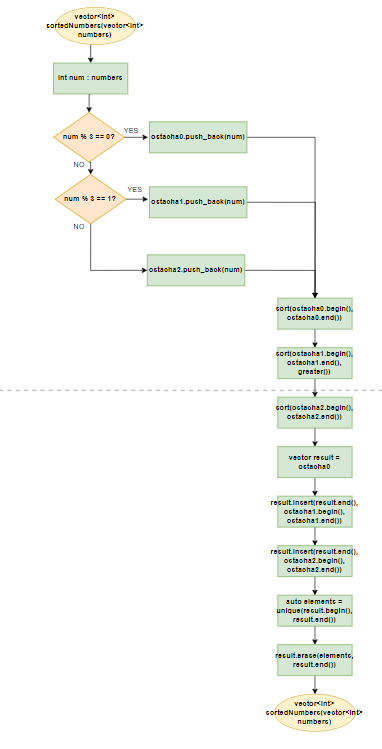
Планований час на виконання – 2 год

Програма №3 VNS Lab 9.

Планований час на виконання – 1,5 год

Програма №4 Algotester Lab 4.





Планований час на виконання – 1,5 год

Програма №5 Algotester Lab 6.

Планований час на виконання – 3 год

Програма №6 Class Practice Task 1.

Планований час на виконання – 40 хв

Програма №7 Class Practice Task 2.

Планований час на виконання – 40 хв

Програма №8 Self-Practice Task.

Планований час на виконання – 40 хв

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

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

Завдання №1

#include <stdio.h>

#include <string.h>

#include <ctype.h>

#include <stdlib.h>

#include <iostream>

void sortDigitsInWord(char word[]) {

int a = strlen(word);

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

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

        if (isdigit(word[j]) && isdigit(word[j + 1]) && word[j] < word[j + 1]) {

            char b = word[j];

            word[j] = word[j + 1];

            word[j + 1] = b;

        }

    }

}

} // Бульбашкове сортування

void sortWordsInString(char str[]) {

    char \*result = strdup(str);

    char \*token;

    token = strtok(result, " ");

    while (token != NULL) {

        sortDigitsInWord(token);

        printf("%s ", token);

        token = strtok(NULL, " ");

    }

    free(result);

}

int main() {

    char s[256];

    gets\_s(s, 256);

    sortWordsInString(s);

    printf("\n");

    return 0;

}

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/718/files#diff-9e0d577d4ea60db1e7c2a0fac28558ab49f0e4aff572cd94a5d56dfe40c60497>

Завдання №2

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

struct TEAM {

    char name[40];

    char city[40];

    int players\_number;

    int points\_number;

};

void addInformation(const char\* filename, int n) {

TEAM\* team = new TEAM[n];

ofstream file(filename);

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

    cout << "Unable to open file!" << endl;

    delete[] team;

    return;

}

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

    cout << "Enter name:" << endl;

    cin.ignore();

    cin.getline(team[i].name, 40);

    cout << "Enter city:" << endl;

    cin.getline(team[i].city, 40);

    cout << "Enter the players number:" << endl;

    cin >> team[i].players\_number;

    cout << "Enter the points number:" << endl;

    cin >> team[i].points\_number;

    file << team[i].name << "\n";

    file << team[i].city << "\n";

    file << team[i].players\_number << "\n";

    file << team[i].points\_number << "\n";

}

file.close();

delete[] team;

}

void showInformation(const char\* filename, int n) {

TEAM\* team = new TEAM[n];

ifstream file(filename);

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

    file.getline(team[i].name, 40);

    file.getline(team[i].city, 40);

    file >> team[i].players\_number;

    file >> team[i].points\_number;

    file.ignore(); // Ignore newline character

    cout << "Name: " << team[i].name << endl;

    cout << "City: " << team[i].city << endl;

    cout << "The players number: " << team[i].players\_number << endl;

    cout << "The points number: " << team[i].points\_number << endl;

}

file.close();

delete[] team;

}

void removeTeamsWithLessPoints(TEAM\*& team, int& n, int minPoints\_number) {

TEAM\* newTeam = new TEAM[n];

int newSize = 0;

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

    if (team[i].points\_number >= minPoints\_number) {

        newTeam[newSize] = team[i];

        newSize++;

    }

}

delete[] team;

team=newTeam;

n=newSize;

}

void addTwoElements(const char\* filename, int k) {

k = 2;

TEAM\* team = new TEAM[k];

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

    cout << "Enter name:" << endl;

    cin.ignore(); // Ignore newline character

    cin.getline(team[i].name, 40);

    cout << "Enter city:" << endl;

    cin.getline(team[i].city, 40);

    cout << "Enter the players number:" << endl;

    cin >> team[i].players\_number;

    cout << "Enter the points number:" << endl;

    cin >> team[i].points\_number;

}

ofstream file(filename, ios::out | ios::app);

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

    cout << "Unable to open file!" << endl;

    delete[] team;

    return;

}

for (int i = k - 1; i >= 0; i--) {

    file.seekp(0, ios::beg); // Move the write pointer to the beginning of the file

    file << "Name: " << team[i].name << "\n";

    file << "City: " << team[i].city << "\n";

    file << "Players Number: " << team[i].players\_number << "\n";

    file << "Points Number: " << team[i].points\_number << "\n\n";

}

file.close();

delete[] team;

}

int main() {

int n;

int minPoints\_number;

cout << "Enter the number of teams:" << endl;

cin >> n;

cout << "Enter the minimal points number:" << endl;

cin >> minPoints\_number;

addInformation("f.dat", n);

showInformation("f.dat", n);

TEAM\* team = new TEAM[n];

ifstream infile("f.dat");

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

    infile.ignore();

    infile.getline(team[i].name, 40);

    infile.ignore();

    infile.getline(team[i].city, 40);

    infile >> team[i].players\_number;

    infile.ignore();

    infile >> team[i].points\_number;

}

infile.close();

removeTeamsWithLessPoints(team, n, minPoints\_number);

addTwoElements("f.dat", 2);

delete[] team;

return 0;

}

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/718/files#diff-dfb9212ac4e396b7894215578e86cb58586ea81086d6e53df3649c3f21cff8ce>

Завдання №3

#include <stdio.h>

#include <string.h>

#include <cctype>

enum FileOpResult { Success, Failure };

FileOpResult processing(char \*file\_from, char \*file\_to, char start\_char, int n1, int n2) {

FILE \*file1 = fopen(file\_from, "r");

FILE \*file2 = fopen(file\_to, "w");

if (file\_from == NULL || file\_to == NULL) {

    return Failure;

}

char line[100];

int current\_line=0;

while (fgets(line, 100, file1) != NULL) {

current\_line++;

if (line[0] == start\_char && current\_line>n1 && current\_line<n2) {

    fputs(line, file2);

}

}

fclose(file1);

fclose(file2);

return Success;

}

int consonants(char\* file\_to) {

FILE\* file2 = fopen(file\_to, "r");

if (file2 == NULL) {

    return -1;

}

char line[100];

int current\_line = 0;

int maxConsonantCount = 0;

int rowWithMaxConsonants = 0;

while (fgets(line, sizeof(line), file2) != NULL) {

    int consonantCount = 0;

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

        if (isalpha(line[i]) && !strchr("AEIOUaeiou", line[i])) {

            consonantCount++;

        }

    }

    if (consonantCount > maxConsonantCount) {

        maxConsonantCount = consonantCount;

        rowWithMaxConsonants = current\_line;

    }

    current\_line++;

}

    fclose(file2);

    return maxConsonantCount;

}

int main() {

char file\_from[] = "file\_from.txt";

char file\_to[] = "file\_to.txt";

char start\_char='A';

FileOpResult result = processing(file\_from, file\_to, start\_char, 2, 4);

if (result == Success) {

    printf("Success\n");

} else if (result == Failure) {

    printf("Error\n");

}

int countedConsonants=consonants(file\_to);

printf("%d", countedConsonants);

return 0;

}

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/718/files#diff-58c63dd706ccb09356d2ad4ddf1d2e537021ace642ef2546e6d4851e2bfb028a>

Завдання №4

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

vector<int> sortedNumbers(vector<int> numbers) {

vector<int> ostacha0, ostacha1, ostacha2;

for (int num : numbers) {

    if (num % 3 == 0) {

        ostacha0.push\_back(num);

    } else if (num % 3 == 1) {

    ostacha1.push\_back(num);

    } else {

    ostacha2.push\_back(num);

    }

}

sort(ostacha0.begin(), ostacha0.end());

sort(ostacha1.begin(), ostacha1.end(), greater<int>());

sort(ostacha2.begin(), ostacha2.end());

vector<int> result = ostacha0;

result.insert(result.end(), ostacha1.begin(), ostacha1.end());

result.insert(result.end(), ostacha2.begin(), ostacha2.end());

auto elements = unique(result.begin(), result.end());

result.erase(elements, result.end());

return result;

}

int main() {

int N;

cin >> N;

vector<int> numbers(N);

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

    cin >> numbers[i];

}

vector<int> Result = sortedNumbers(numbers);

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

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

    cout << Result[i] << " ";

}

return 0;

}

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/718/files#diff-48e59693e236202a0e26b4b5dd96f4a79ee9e91924c5407f6151bed4cf9c0355>

Завдання №5

#include <iostream>

#include <vector>

#include <set>

#include <algorithm>

using namespace std;

void process(vector<vector<int>> field, int x, int y) {

if (field[x-1][y-1]==0) {

    set<int> numbers;

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

        numbers.insert(i);

    }

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

        numbers.erase(field[i][y-1]);

        numbers.erase(field[x-1][i]);

    }

    cout << numbers.size() << endl;

    for (int a : numbers) {

        cout << a << " ";

    }

} else {

    cout << "1" << endl << field[x-1][y-1];

}

cout << endl;

cout << endl;

}

int main() {

int N, Q;

cin >> N;

const int rows=N;

const int cols=N;

vector<vector<int>> field;

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

    vector<int> rowOfField(cols);

    for (int j = 0; j < cols; ++j) {

        char number;

        cin >> number;

        rowOfField[j] = number - '0';

    }

    field.push\_back(rowOfField);

}

cin >> Q;

vector<pair<int, int>> coordinates;

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

    int x, y;

    cin >> x >> y;

    coordinates.emplace\_back(x, y);

}

for (auto coordinate : coordinates) {

    process(field, coordinate.first, coordinate.second);

}

return 0;

}

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/718/files#diff-dba51fe7912ce9d2073dc55f3fb8abe230ee92063e589462939f90e05de15994>

Завдання №6

#include <stdio.h>

#include <string.h>

enum FileOpResult { Success, Failure };

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

    FILE \*file = fopen(name, "w");

    if (file == NULL) {

        return Failure;

    }

    fprintf(file, "%s", content);

    fclose(file);

    return Success;

}

int main() {

    char name[] = "MyNewFile.txt";

    char content[100];

    fgets(content, sizeof(content), stdin);

    FileOpResult result = write\_to\_file(name, content);

    if (result == Success) {

        printf("Success\n");

    } else if (result == Failure) {

        printf("Error\n");

    }

    return 0;

}

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/718/files#diff-0a5ff0090726d13f3132edf77377f07dc147a698c4324100a5b5ca0000924a2c>

Завдання №7

#include <stdio.h>

#include <string.h>

enum FileOpResult { Success, Failure };

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

FILE \*file1 = fopen(file\_from, "r");

FILE \*file2 = fopen(file\_to, "w");

if (file\_from == NULL || file\_to == NULL) {

    return Failure;

}

int process;

while ((process = fgetc(file1)) != EOF) {

    fputc(process, file2);

}

fclose(file1);

fclose(file2);

return Success;

}

int main() {

char file\_from[] = "file\_from.txt";

char file\_to[] = "file\_to.txt";

FileOpResult result = copy\_file(file\_from, file\_to);

if (result == Success) {

    printf("Success\n");

} else if (result == Failure) {

    printf("Error\n");

}

return 0;

}

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/718/files#diff-4e364a1be94d207ee5ba49b07cd8668a4004ac296e8740a25a04a3e43b354f5c>

Завдання №8

#include <iostream>

using namespace std;

int main() {

int n, k1=15, k, p, z;

cin>>n;

char \*line=new char[n];

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

    cin >> line[i];

}

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

if (line[i] == 'k' ) {

    k1=max(k1-1, 0);

}

else if (line[i] == 'p' ) {

    if (k1!=0) {

        k1=15;

    } else {

        k1=0;

    }

}

else if (line[i] == 'z' ) {

    if (k1!=0) {

        k1=0;

    } else {

        k1=15;

    }

}

}

cout << k1 << endl;

delete[] line;

return 0;

}

<https://github.com/artificial-intelligence-department/ai_programming_playground/pull/718/files#diff-b544b182547fbed2635b65bcd87c2bb11b0fc0dcd54b238e7f87870f8fd03e3a>

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

Завдання №1

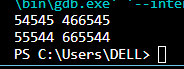


Рисунок 9. Результат виконання програми 1.

Час затрачений на виконання завдання – 2 год

Завдання №2



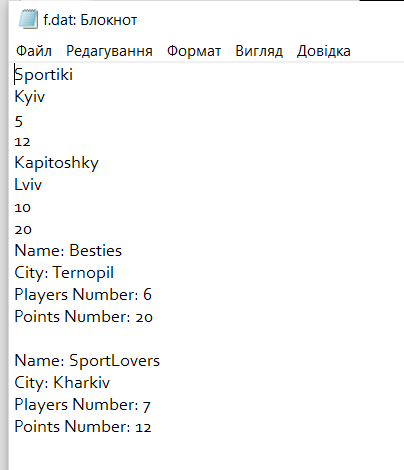
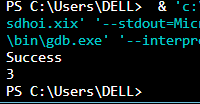
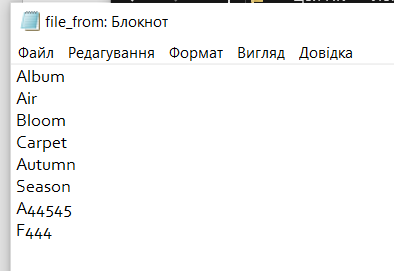


Рисунок 10. Результат виконання програми 2.

Час затрачений на виконання завдання – 2 год

Завдання №3





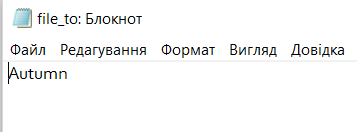


Рисунок 11. Результат виконання програми 3.

Час затрачений на виконання завдання – 2 год

Завдання №4

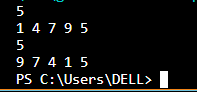


Рисунок 12. Результат виконання програми 4.

Час затрачений на виконання завдання – 2 год

Завдання №5

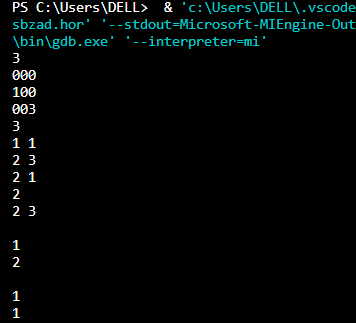
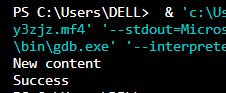


Рисунок 13. Результат виконання програми 5.

Час затрачений на виконання завдання – 3 год

Завдання №6



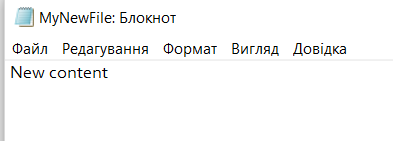


Рисунок 14. Результат виконання програми 6.

Час затрачений на виконання завдання – 1,5 год

Завдання №7

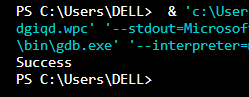


Рисунок 15. Результат виконання програми 7.

Час затрачений на виконання завдання – 1,5 год

Завдання №8

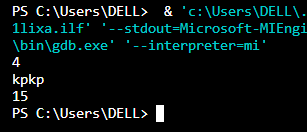
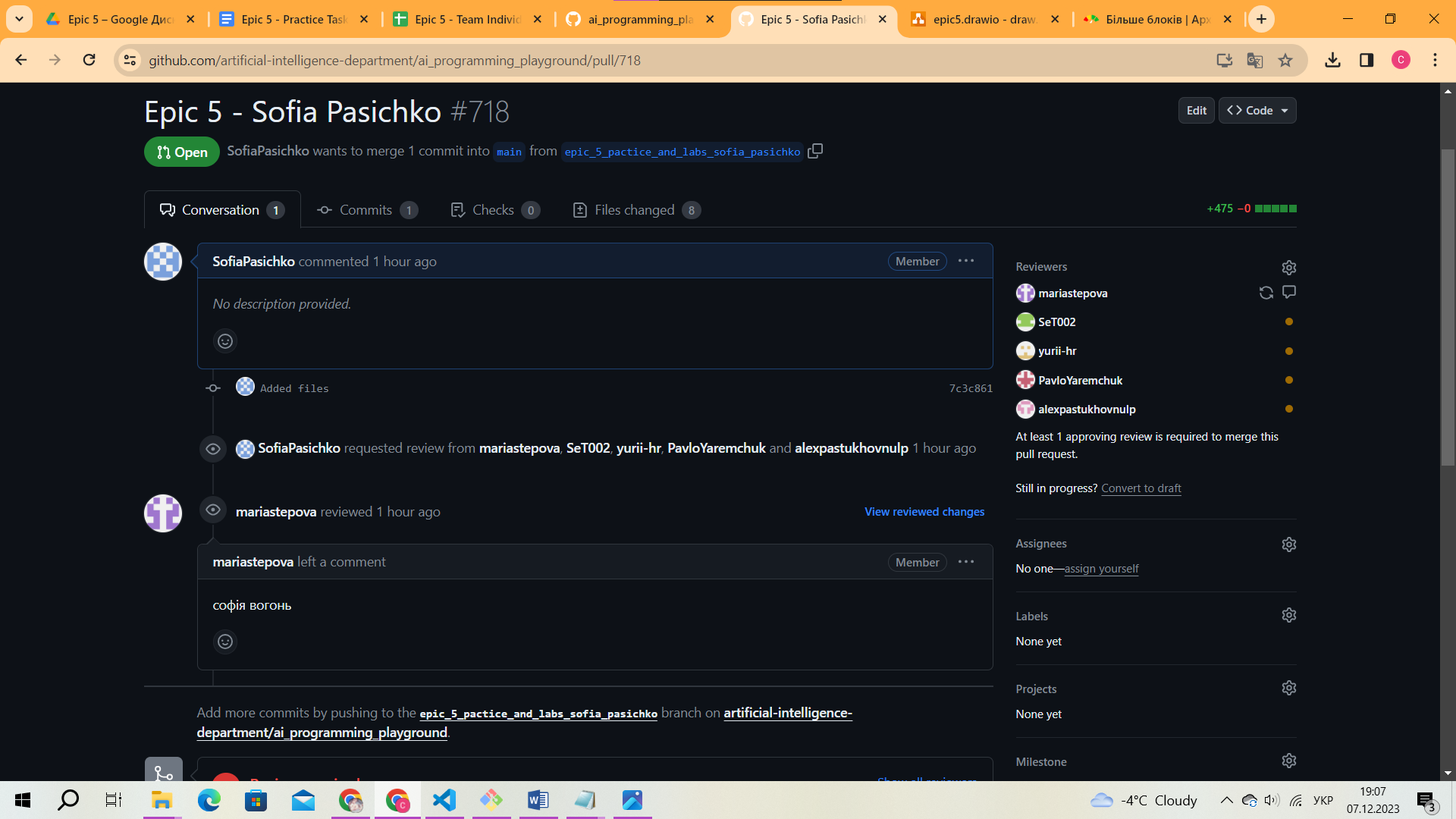


Рисунок 16. Результат виконання програми 8.

Час затрачений на виконання завдання – 40 хв

## **6. Кооперація з командою:**





# **Висновки:**

Протягом епіку 5 ознайомилася з роботою з файлами. Повторила системи числення. Вивчила різницю та особливості роботи з текстовими та бінарними файлами. Ознайомилася зі створенням та використанням бібліотек.