Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**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(ла):***

Cтудент групи ШІ-14

Грицишин Максим Іванович

# **Тема роботи:**

# Ознайомлення з файлами, системами числення, бінарними файлами, символами та рядковими змінними та текстовими файлами, стандартною бібліотекою та методами роботи з файлами, створенням і використанням бібліотек. Практичне застосування знань на основі виконання лабораторних та практичних робіт.

# **Мета роботи:**

* Theory Education Activities
* Requirements management (understand tasks) and design activities (draw flow diagrams and estimate tasks 3-9)
* Lab# programming: VNS Lab 6
* Lab# programming: VNS Lab 8
* Lab# programming: VNS Lab 9
* Lab# programming: Algotester Lab 4
* Lab# programming: Algotester Lab 6
* Practice# programming: Class Practice Task
* Practice# programming:  Self Practice Task
* Result Documentation Report and Outcomes Placement Activities (Docs and Programs on GitHub)

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Файли. Бінарні файли. Текстові файли. Методи роботи з файлами
* Тема №2: Системи числення
* Тема №3: Символи. Рядкові змінні
* Тема №4: Стандартна бібліотека

1. Індивідуальний план опрацювання теорії:

* Тема №1: Файли. Бінарні файли. Текстові файли. Методи роботи з файлами
  + Джерела Інформації
    - <https://acode.com.ua/urok-220-bazovyj-fajlovyj-vvid-i-vyvid/>
    - Ресурси в інтернеті
    - Відео на Youtube
    - ChatGPT
  + Що опрацьовано:
    - Ознайомлено з бібліотекою, яку потрібно використовувати при роботі з файлами, зокрема
    - Використано у лабораторних та практичній роботах

ChatGPT

* + Статус: Ознайомлений
  + Початок опрацювання теми: 03.12.2023
  + Звершення опрацювання теми: 10.12.2023
* Тема №2: Системи числення
  + Джерела Інформації:
    - <https://acode.com.ua/urok-47-konvertatsiya-chysel-z-dvijkovoyi-systemy-chyslennya-v-desyatkovu-i-navpaky/>
    - Лекції з дисципліни “Архітектура комп’ютера”
    - Лекції та пратичні пари з дисципліни “Мови та парадигми програмування”
  + Що опрацьовано:
    - Ознайомлено теоретично
    - Кілька вдалих спроб переводу з двійкової в десяткову і навпаки
  + Статус: Ознайомлений частково
  + Початок опрацювання теми: 15.09.2023
  + Завершення опрацювання теми: --
* Тема №3: Символи. Рядкові змінні
  + Джерела Інформації:
    - <https://acode.com.ua/urok-38-symvolnyj-typ-danyh-char/>
    - <https://studfile.net/preview/3908531/>
    - Інтернет-ресурси
    - ChatGPT
  + Що опрацьовано:
    - Доволі часто використовується у роботах з попередніх епіків
    - Використано у лабораторних та практичній роботах
  + Статус: Ознайомлений
  + Початок опрацювання теми: 30.10.2023
  + Завершення опрацювання теми: 08.12.2023
* Тема №4: Стандартна бібліотека
  + Джерела Інформації:
    - <https://acode.com.ua/urok-205-kontejnery-stl/>
    - <https://acode.com.ua/urok-207-algorytmy-stl/>
    - <https://acode.com.ua/urok-206-iteratory-stl>
    - ChatGPT
  + Що опрацьовано:
    - Ознайомлено
  + Статус: Ознайомлений частково
  + Початок опрацювання теми: 03.12.2023
  + Завершення опрацювання теми: —

# **Виконання роботи:**

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

Завдання №1 Лабораторна робота №6 "Рядки"

* Варіант завдання: 6
* 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. Перетворити рядок так, щоб всі букви в ньому були відсортовані за зростанням.
* Важливі деталі для врахування в імплементації програми: --

Завдання №2 Лабораторна робота №8 "Блоковий ввід-вивід"

* Варіант завдання: 6
* 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 "Школяр":

- прізвище, ім'я, по батькові;

- клас;

- номер телефону;

- оцінки по предметах (математика, фізика, українська мова, література).

Знищити всі елементи, у яких є 2 хоча б з одного предмету, додати елемент у

початок файлу.

* Важливі деталі для врахування в імплементації програми: --

Завдання №3 Лабораторна робота №9 "Ввід-вивід рядків"

* Варіант завдання: 6
* Деталі завдання: Створити текстовий файл F1 не менше, ніж з 10 рядків і записати в нього інформацію.

1) Скопіювати з файлу F1 у файл F2 рядки, починаючи з N до K.

2) Підрахувати кількість приголосних букв у файлі F2.

* Важливі деталі для врахування в імплементації програми: --

Завдання №4 Лабораторна робота №4 у Algotester

* Варіант завдання: 1
* Деталі завдання: Дано 2 цiлих чисел масиви, розмiром N та M.

Завдання вивести:

1. Рiзницю N-M

2. Рiзницю M-N

3. Їх перетин

4. Їх об’єднання

5. Їх симетричну рiзницю

* Важливі деталі для врахування в імплементації програми:

Вхiднi данi:

У першому рядку цiле число N - розмiр масиву 1

У другому рядку N цiлих чисел - елементи масиву 1

У третьому рядку цiле число M - розмiр масиву 2

У четвертом рядку M цiлих чисел - елементи масиву 2

Вихiднi данi:

Вивести результат виконання 5 вищезазначених операцiй у форматi:

У першому рядку цiле число N - розмiр множини

У наступному рядку N цiлих чисел - посортована у порядку зростання множина

Завдання №5 Лабораторна робота №6 у Algotester

* Варіант завдання: 1
* Деталі завдання: Дано N слiв та число K. Завдання перечислити букви в словах, якi зустрiчаються в текстi бiльше-рiвне нiж K разiв (саме слово, не буква!). Великi та маленькi букви вважаються однаковими, виводити необхiдно малi, посортованi вiд останньої до першої у алфавiтi. Букву потрiбно виводити лише один раз. У випадку якщо таких букв немає - вивести "Empty!".
* Важливі деталі для врахування в імплементації програми:

Вхiднi данi:

Цiлi числа N та K - загальна кiлькiсть слiв та мiнiмальна кiлькiсть слiв щоб враховувати

букви цього слова в результатi.

N стрiчок s

Вихiднi данi:

У першому рядку цiле число M - кiлькiсть унiкальних букв

У другому рядку унiкальнi букви через пробiли

Завдання №6 Практична робота “Робота з текстовими файлами”

1. Задача №1 – Запис текстової стрічки у файл із заданим ім’ям

* Деталі завдання:

1. створити файл із заданим ім’ям; якщо файл існує – перезаписати його вміст
2. написати код стійкий до різних варіантів вхідних параметрів
3. name – ім’я, може не включати шлях
4. записати у файл вміст стрічки content, прочитати content із стандартного вводу
5. повернути статус операції: Success – все пройшло успішно, Failure – файл не вдалося створити, або збій операції відкриття, запису даних, чи закриття файла.

* Важливі деталі для врахування в імплементації програми: Реалізувати функцію створення файла і запису в нього даних:

enum FileOpResult { Success, Failure, … };

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

1. Задача №2 – Копіювання вмісту файла у інший файл

* Деталі завдання:

1. 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
2. file\_from, file\_to – можуть бути повним або відносним шляхом
3. повернути статус операції: Success – все пройшло успішно, Failure – файл не вдалося створити, або збій операції відкриття, читання чи запису даних, закриття файла.

* Важливі деталі для врахування в імплементації програми: Реалізувати функцію створення файла і запису в нього даних:

enum FileOpResult { Success, Failure, … };

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

Завдання №7 Власна практична робота “Лабораторна робота №4 у Algotester”

* Варіант завдання: 3
* Деталі завдання: Дано масив, який складається з N додатніх цiлих чисел. Завдання - роздiлити його на три частини, за остачею від ділення на 3, за зростанням остачi (спочатку йдуть числа, у яких остача 0, потім числа з остачею 1 i тодi числа з остачею Далi необхiдно тi елементи, остача вiд дiлення на 3 яких парна, посортувати за зростанням, а тi, у яких остача 1, - за спаданням. Пiсля цього видалiть усi дублiкати з масиву. Виведiть остаточний масив.
* Важливі деталі для врахування в імплементації програми:

Вхiднi данi:

У першому рядку N - кiлькiсть чисел.

У другому рядку N чисел ai - елементи масиву.

Вихiднi данi:

У першому рядку M - кiлькiсть чисел у масиву

У другому рядку M посоротованих за умовою чисел.

Завдання №8 Власна практична робота “Хелловін”

* Деталі завдання: У Зеника і Марічки є по одному мішку цукерок. Усього в мішку Зеника є n цукерок, причому i-та із них коштує ai гривень. Аналогічно, у мішку Марічки є m цукерок, і ціна i-ї рівна bi гривень. Вони хочуть дати Андрію дві цукерки — одну з мішка Зеника, а іншу — з мішка Марічки. Будучи доволі жадібними, воно готові віддати Андрію лише найдешевші цукерки. Яка сумарна вартість двох цукерок, які отримає Андрій?
* Важливі деталі для врахування в імплементації програми:

Вхiднi данi:

У першому рядку задано два цілих числа n та m — кількості цукерок в мішках Зеника та Марічки відповідно. У другому рядку задано n цілих чисел, розділених пробілами. Вони описують вартості Зеникових цукерок. У третьому рядку аналогічно описані m цукерок Марічки.

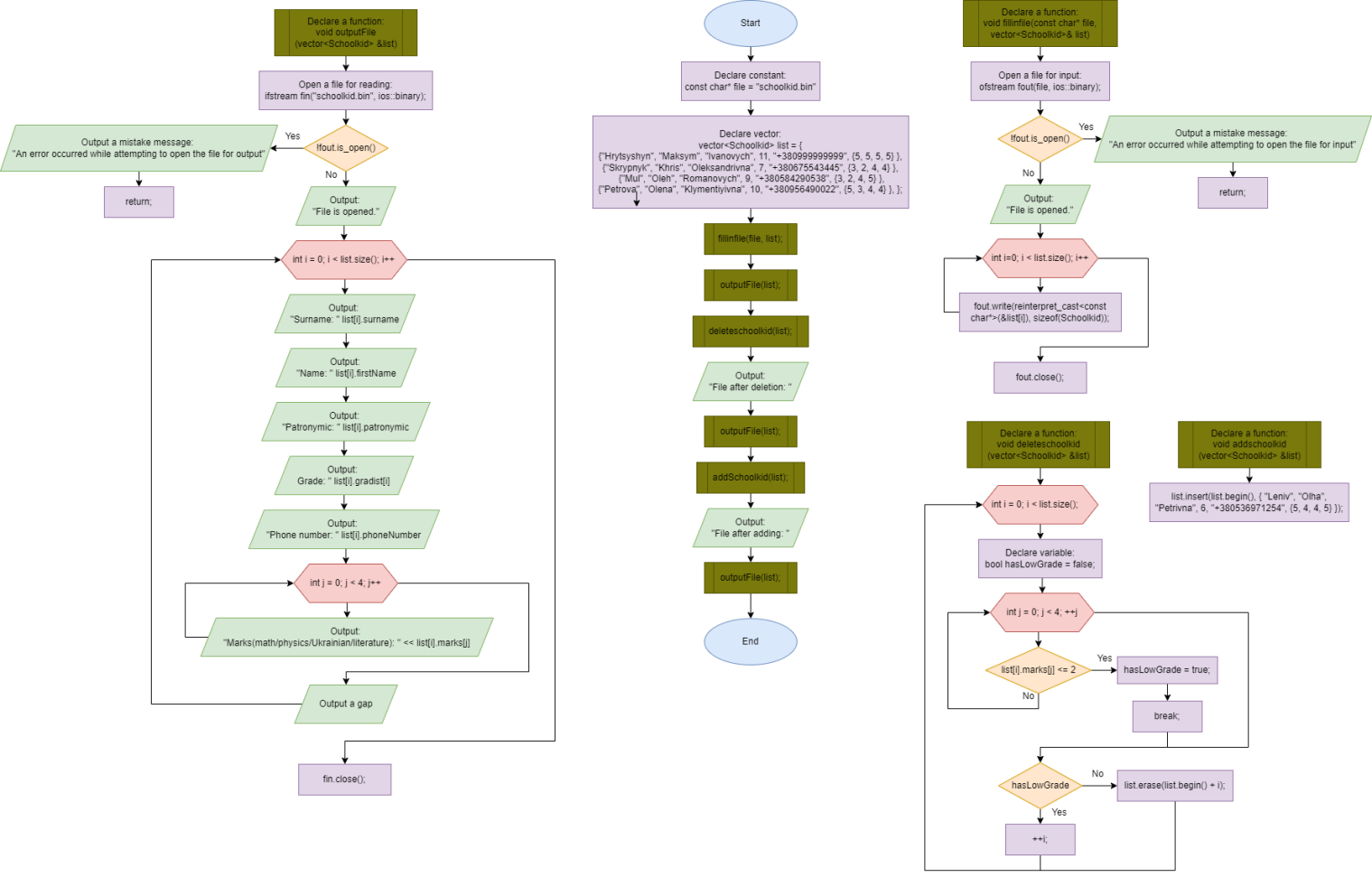
Вихiднi данi:

У єдиному рядку виведіть одне ціле число — вартість подарунку, який отримає Андрій.

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

Програма №1 Лабораторна робота №6 "Рядки"

* Планований час на реалізацію: 2год
* Важливі деталі для врахування в імплементації: У рядку можуть бути пробіли. Кількість символів обмежена (255).

Програма №2 Лабораторна робота №8 "Блоковий ввід-вивід"

Малюнок : Блок-схема до програми №2

* Планований час на реалізацію: 4год
* Важливі деталі для врахування в імплементації: для кожної «дії» варто використовувати окрему функцію

Програма №3 Лабораторна робота №9 "Ввід-вивід рядків"

* Планований час на реалізацію: 3год
* Важливі деталі для врахування в імплементації: файл F1 має складати не менше, ніж з 10 рядків

Програма №4 Лабораторна робота №4 у Algotester

* Планований час на реалізацію: 3год
* Важливі деталі для врахування в імплементації: щоб працювати із засобами STL, потрібно посортувати рядок

Програма №5 Лабораторна робота №6 у Algotester

* Планований час на реалізацію: 2год
* Важливі деталі для врахування в імплементації: для оптимізації варто використовувати unordered\_map та set з їхніми однойменними бібліотеками.

Програма №6 Практична робота “Робота з текстовими файлами”

* Планований час на реалізацію: 4год
* Важливі деталі для врахування в імплементації: потрібно використати вказані в умові засоби

Програма №7 Власна практична робота “Лабораторна робота №4 у Algotester”

* Планований час на реалізацію: 2год
* Важливі деталі для врахування в імплементації: напочатку вектор варто розділити на 3Б за принципом остачі, і об’єднати лише після сортування

Програма №8 Власна практична робота “Хелловін”

* Планований час на реалізацію: 40хв
* Важливі деталі для врахування в імплементації: --

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

*Конфігурація середовища не змінилась.*

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

Завдання №1 Лабораторна робота №6 "Рядки"

#include <iostream>

#include <algorithm>

#include <cstring>

#include <cctype>

using namespace std;

const int amount = 255;

int main() {

    char str[amount];

    cout << "Enter a string: ";

    cin.getline(str, amount, '.');

    string line;

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

        if (isalpha(str[i])){

           line += tolower(str[i]);

        }

    }

    sort(line.begin(), line.end());

    cout << "Updated string: " << line << endl;

}

Деталі програми: використано getline() (бібліотека cstring) для введення рядка, який можна зчитувати незважаючи на пробіли, а в кінці обов’язково стоїть крапка. Задля того, щоб всі літери в рядку звести до малих, використовується tolower(), а щоб перед тим перевірити чи всі символи рядка є літерами використовується isalpha() (обидва – бібліотека cctype). Щоб посортувати букви, використовувалися sort() та оператори .begin() і .end().

Покликання на пул-реквест: <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/826/files#diff-094eb5afed61b05ccd94d04dadeed7ac0b7c6263549c60990d1d14c4ca0b492a>

Завдання №2 Лабораторна робота №8 "Блоковий ввід-вивід"

#include <iostream>

#include <vector>

#include <fstream>

using namespace std;

struct Schoolkid {

    string surname;

    string firstName;

    string patronymic;

    int grade;

    string phoneNumber;

    int marks[4];

};

void fillinfile(const char\* file, vector<Schoolkid>& list) {

    ofstream fout(file, ios::binary);

    if (!fout.is\_open()) {

        cerr << "An error occurred while attempting to open the file for input" << endl;

        return;

    } else {

        cout << "File is opened." << endl;

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

            fout.write(reinterpret\_cast<const char\*>(&list[i]), sizeof(Schoolkid));

        }

    }

    fout.close();

}

void deleteschoolkid(vector<Schoolkid> &list) {

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

        bool hasLowGrade = false;

        for (int j = 0; j < 4; ++j) {

            if (list[i].marks[j] <= 2) {

                hasLowGrade = true;

                break;

            }

        }

        if (hasLowGrade) {

            list.erase(list.begin() + i);

        } else {

            ++i;

        }

    }

}

void addSchoolkid(vector<Schoolkid> &list) {

    list.insert(list.begin(), { "Leniv", "Olha", "Petrivna", 6, "+380536971254", {5, 4, 4, 5} });

}

void outputFile(vector<Schoolkid> &list) {

    ifstream fin("schoolkid.bin", ios::binary);

    if (!fin.is\_open()) {

        cerr << "An error occurred while attempting to open the file for output" << endl;

        return;

    }

    cout << "File is opened." << endl;

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

        cout << "Surname: " << list[i].surname << endl;

        cout << "Name: " << list[i].firstName << endl;

        cout << "Patronymic: " << list[i].patronymic << endl;

        cout << "Grade: " << list[i].grade << endl;

        cout << "Phone number: " << list[i].phoneNumber << endl;

        for(int j = 0; j < 4; j++) {

            cout << "Marks(math/physics/Ukrainian/literature): " << list[i].marks[j] << endl;

        }

        cout << endl;

    }

    fin.close();

}

int main() {

    const char\* file = "schoolkid.bin";

    vector<Schoolkid> list = {

    {"Hrytsyshyn", "Maksym", "Ivanovych", 11, "+380999999999", {5, 5, 5, 5} },

    {"Skrypnyk", "Khris", "Oleksandrivna", 7, "+380675543445", {3, 2, 4, 4} },

    {"Mul", "Oleh", "Romanovych", 9, "+380584290538", {3, 2, 4, 5} },

    {"Petrova", "Olena", "Klymentiyivna", 10, "+380956490022", {5, 3, 4, 4} },

    };

    fillinfile(file, list);

    outputFile(list);

    deleteschoolkid(list);

    cout << "File after deletion: " << endl;

    outputFile(list);

    addSchoolkid(list);

    cout << "File after adding: " << endl;

    outputFile(list);

    return 0;

}

Деталі програми: використано структуру згідно зі завданням, класи ifstream та ofstream бібліотеки fstream для відкриття файлів з певною метою, методи .open() та .close(), прапор binary. Також використано вектор для заповнення даними структури та подальшої роботи з нею.

Покликання на пул-реквест: <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/826/files#diff-3f7a6eed28e72de58a20c7b4ede067608cebf18145d7b467dd504919d48623ad>

Завдання №3 Лабораторна робота №9 "Ввід-вивід рядків"

#include <iostream>

#include <string>

#include <fstream>

#include <vector>

using namespace std;

int main() {

    ofstream F1("F1.txt");

    if (!F1.is\_open()) {

        cerr << "An error occurred while attempting to open the file" << endl;

        return 1;

    }

    vector<string> str(10);

    for (int a=0; a<10; ++a) {

        cout << "Enter a string: ";

        getline(cin, str[a]);

        F1 << str[a] << endl;

        cin.ignore();

    }

    F1.close();

    int N=0;

    int K=0;

    do{

        cout << "Enter the number of the line starting from which copying to another file will take place: ";

        cin >> N;

    } while (N <= 0 || N >= 10);

    do{

        cout << "Enter the line number that will end copying to another file: ";

        cin >> K;

    } while (K <= 0 || K >= 10);

    ifstream F1read("F1.txt");

    ofstream F2("F2.txt");

    if (!F1read.is\_open() || !F2.is\_open()) {

        cerr << "An error occurred while attempting to open file F1 or F2" << endl;

        return 1;

    }

    string line;

    int a=1;

    for(int a=1; a<=K || a >= N && getline(F1read, line); ++a) {

        F2 << "\n" << line << endl;

    }

    F1read.close();

    F2.close();

    ifstream F2read("F2.txt");

    if (!F2read.is\_open()) {

        cerr << "An error occurred while attempting to open the file" << endl;

        return 1;

    }

    int consonant;

    char symbol;

    while(F2read.get(symbol)) {

        if (isalpha(symbol) && (symbol != 'a' || symbol != 'A' || symbol != 'e' || symbol != 'E' || symbol != 'i' || symbol != 'I' || symbol != 'o' || symbol != 'O' || symbol != 'u' || symbol != 'U' || symbol != 'y' || symbol != 'Y')) {

            ++consonant;

        }

    }

    F2read.close();

    cout << "The amount of consonants in F2.txt: " << consonant << endl;

    return 0;

}

Деталі програми: Використано класи ifstream та ofstream для створення файлів F1 та F2 і копіювання вибраного контенту із файлу F1 у файл F2, а також потрібні для роботи з ними методи, прапори, класи та оператори. Використано вектор для роботи з кількома рядками, а також .get() та isaplpha().

Покликання на пул-реквест: <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/826/files#diff-00231e3fd43c01e5541ff6af5096c741baa15f2dfdb4fc7d43e7684f2147737e>

Завдання №4 Лабораторна робота №4 у Algotester

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

int main() {

    int N, M;

    do {

        cin >> N;

    } while (N<1 || N>100);

    vector<int> arr1(N);

    for(int a=0; a<N; a++) {

        cin >> arr1[a];

    }

    do {

        cin >> M;

    } while (M<1 || M>100);

    vector<int> arr2(M);

    for(int a=0; a<M; a++) {

        cin >> arr2[a];

    }

    vector<int> differenceNM;

    vector<int> differenceMN;

    vector<int> intersection;

    vector<int> unification;

    vector<int> symdifference;

    sort(arr1.begin(), arr1.end());

    sort(arr2.begin(), arr2.end());

    set\_difference(arr1.begin(), arr1.end(), arr2.begin(), arr2.end(), back\_inserter(differenceNM));

    set\_difference(arr2.begin(), arr2.end(), arr1.begin(), arr1.end(), back\_inserter(differenceMN));

    set\_intersection(arr1.begin(), arr1.end(), arr2.begin(), arr2.end(), back\_inserter(intersection));

    set\_union(arr1.begin(), arr1.end(), arr2.begin(), arr2.end(), back\_inserter(unification));

    set\_symmetric\_difference(arr1.begin(), arr1.end(), arr2.begin(), arr2.end(), back\_inserter(symdifference));

    cout << "\n" << differenceNM.size() << endl;

    for(int element : differenceNM) {

        cout << element << " ";

    }

    cout << endl;

    cout << "\n" << differenceMN.size() << endl;

    for(int element : differenceMN) {

        cout << element << " ";

    }

    cout << endl;

    cout << "\n" << intersection.size() << endl;

    for(int element : intersection) {

        cout << element << " ";

    }

    cout << endl;

    cout << "\n" << unification.size() << endl;

    for(int element : unification) {

        cout << element << " ";

    }

    cout << endl;

    cout << "\n" << symdifference.size() << endl;

    for(int element : symdifference) {

        cout << element << " ";

    }

    return 0;

}

Деталі програми: Використано вектори для запису в них масивів та вектори з результатами операцій над ними. Застосовано **засоби STL (std::set\_intersection, std::set\_symmetric\_difference, std::set\_difference, std::set\_union) та функція sort з потрібними операторами begin() та end().**

Покликання на пул-реквест: <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/826/files#diff-44b91e971c574bbaa223d0fc75981553d46bc789d7040cbfd9af990785a084da>

Завдання №5 Лабораторна робота №6 у Algotester

#include <iostream>

#include <unordered\_map>

#include <vector>

#include <set>

#include <algorithm>

using namespace std;

int main() {

    int N, K;

    cin >> N >> K;

    vector<string> s(N);

    unordered\_map<string, int> amount;

    for (int a=0; a<N; ++a) {

        cin >> s[a];

        transform(s[a].begin(), s[a].end(), s[a].begin(), [](unsigned char letter) { return tolower(letter); });

        ++amount[s[a]];

    }

    set<char, greater<char>> uniques;

    for(const pair<const string, int>&eachmeet : amount) {

        if(eachmeet.second >= K) {

            for (char letter : eachmeet.first) {

                uniques.insert(letter);

        }

    }

    }

    int M = uniques.size();

    if (!uniques.empty()) {

        cout << M << endl;

        for (auto letters : uniques) {

            cout << letters << " ";

        }

    } else {

        cout << "Empty!" << endl;

    }

    return 0;

}

Деталі програми: У цьому коді unordered\_map використовується, аби зберегти кількість входжень рядка s (рядок як ключ, ціле число як значення). set використовується для зворотньо(за допомогою greater<char>) впорядкованого зберігання унікальних літер рядка. Перед додаванням рядків до unordered\_map функція transform перетворює всі символи рядків у малий регістр.

Покликання на пул-реквест: <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/826/files#diff-2be0b9c4f718b8aec059e8eccb22cf22c02ff0a28584f91f8bbed098ce65fa57>

Завдання №6 Практична робота “Робота з текстовими файлами”

#include <iostream>

#include <fstream>

using namespace std;

enum FileOpResult {Success, Failure};

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

    ofstream fout(name, ios::out);

    if (!fout.is\_open()) {

        return Failure;

    }

    fout << content;

    fout.close();

    return Success;

}

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

    ifstream fin(file\_from, ios::in);

    if (!fin.is\_open()) {

        return Failure;

    }

    ofstream fout(file\_to, ios::out);

    if (!fout.is\_open()) {

        return Failure;

    }

    char ch;

    while (fin.get(ch)) {

        fout << ch;

    }

    fin.close();

    fout.close();

    return Success;

}

int main() {

    char name[50];

    cout << "Enter a file name: ";

    cin >> name;

    char content[250];

    cin.ignore();

    cout << "Enter a file content: ";

    cin.getline(content, 250);

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

    if (writingOutcome == Success) {

        cout << "The file was successfully written to" << endl;

    } else {

        cout << "An error occurred while attempting to write to the file" << endl;

    }

    char file\_to[50];

    cout << "Enter a file name to copy to: ";

    cin >> file\_to;

    FileOpResult copyingOutcome = copy\_file(name, file\_to);

    if (copyingOutcome == Success) {

        cout << "The file was successfully copied" << endl;

    } else {

        cout << "An error occurred while attempting to copy the file" << endl;

    }

    return 0;

}

Деталі програми: У цьому коді використовується enum для оголошення двох константних значень, що робить код більш зрозумілим і полегшує розуміння того, що повертає функція. Використовується бібліотека fstream для роботи з файлами.

Покликання на пул-реквест: <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/826/files#diff-87bf895108bbb8e2e5b7558562ae824aef3ac9ce71945c3094b5a8eaeecddf62>

Завдання №7 Власна практична робота “Лабораторна робота №4 у Algotester”

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

int main() {

    int N;

    cin >> N;

    vector<int> a(N);

    for (int i = 0; i < N; i++) {

        cin >> a[i];

    }

    vector<int> remainder0, remainder1, remainder2;

    for (int i=0; i<N; i++) {

        if (a[i] % 3 == 0) {

            remainder0.push\_back(a[i]);

        } else if (a[i] % 3 == 1) {

            remainder1.push\_back(a[i]);

        } else {

            remainder2.push\_back(a[i]);

        }

    }

    sort(remainder0.begin(), remainder0.end());

    sort(remainder1.begin(), remainder1.end(), greater<int>());

    sort(remainder2.begin(), remainder2.end());

    vector<int> anew;

    anew.insert(anew.end(), remainder0.begin(), remainder0.end());

    anew.insert(anew.end(), remainder1.begin(), remainder1.end());

    anew.insert(anew.end(), remainder2.begin(), remainder2.end());

    auto forerase = unique(anew.begin(), anew.end());

    anew.erase(forerase, anew.end());

    cout << anew.size() << endl;

    for (int each : anew) {

        cout << each << " ";

    }

    cout << endl;

    return 0;

}

Деталі програми: Спочатку початковий вектор поділено на 3 вектори за остачею 0,1,2. Далі за допомогою sort() всі три вектори посортовано, проте вектор remainder2 завдяки greater<int> посортовано в зворотному порядку. Далі за допомогою insert 3 вектори об’єднано в один. Потім використовується unique, а після цього erase, щоб отримати вектор лише з унікальними значеннями.

Покликання на пул-реквест: <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/826/files#diff-dc2c5baf820dcecc5799c1cabcb1ee97caf870a542cc8c79b45b0b9e65d32cbc>

Завдання №8 Власна практична робота “Хелловін”

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

int main() {

    int n , m;

    cin >> n >> m;

    vector<int> a(n);

    vector<int> b(m);

    for(int i=0; i<n; i++) {

        cin >> a[i];

    }

    for(int i=0; i<m; i++) {

        cin >> b[i];

    }

    sort(a.begin(), a.end());

    sort(b.begin(), b.end());

    auto amin = a.begin();

    auto bmin = b.begin();

    int min = \*amin + \*bmin;

    cout << min << endl;

}

Деталі програми: Використано вектори, що запонуються значеннями. За допомогою функції sort() та операторів begin і end знайдено мінімальні значення у векторах, а потім і їх суму.

Покликання на пул-реквест: <https://github.com/artificial-intelligence-department/ai_programming_playground/pull/826/files#diff-50378a7b5736228a167a78ed70dd2b3ab6e591ae65fd36b703371f8164007d6a>

Покликання на умову завдання: <https://algotester.com/uk/ArchiveProblem/DisplayWithEditor/40475>

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

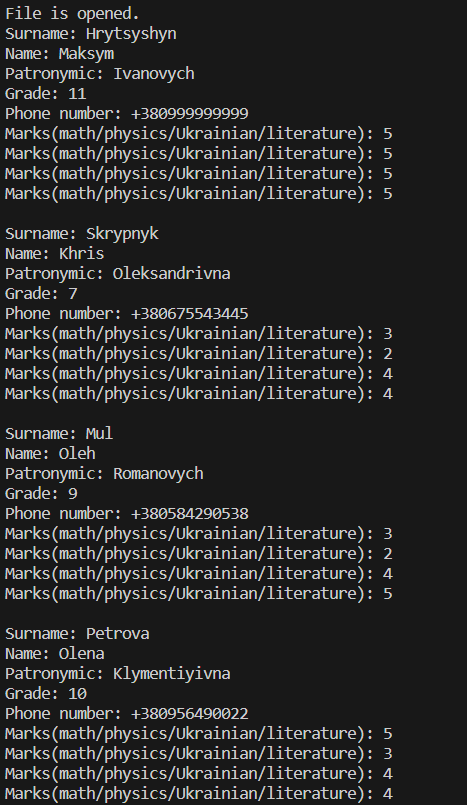
Завдання №1 Лабораторна робота №6 "Рядки"



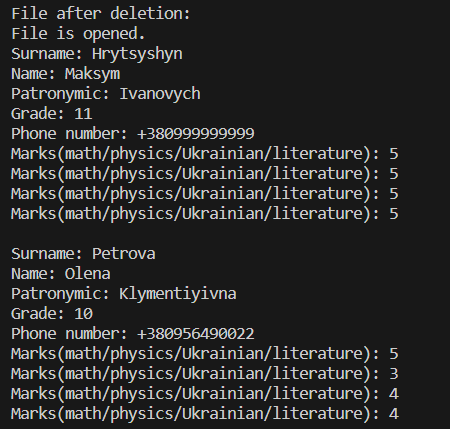
Малюнок 2: Успішна компіляція програми із завдання №1

Час затрачений на виконання завдання: 1.5год

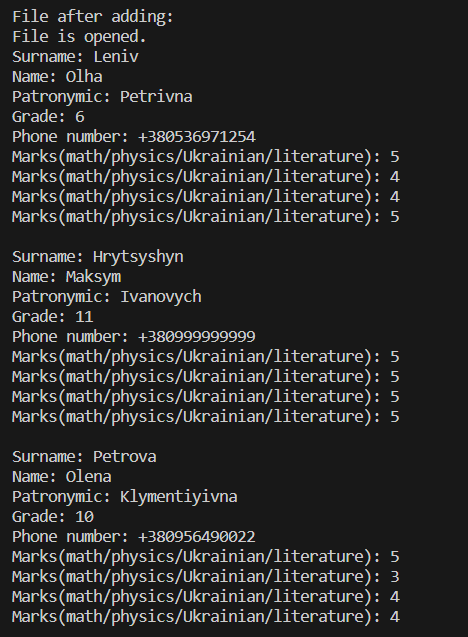
Завдання №2 Лабораторна робота №8 "Блоковий ввід-вивід"



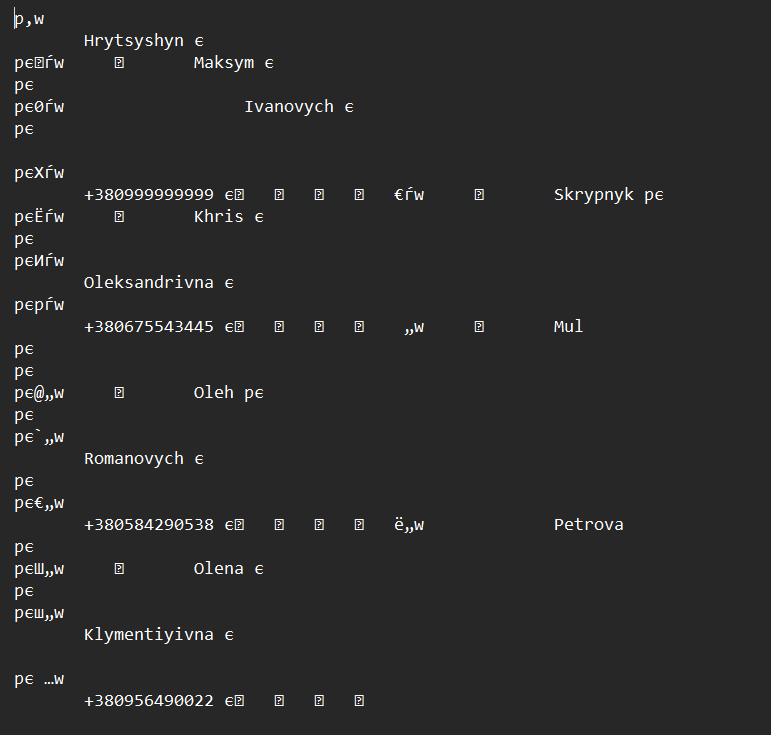
Малюнок 3: Виведення початкового списку школярів



Малюнок 4: Виведення списку школярів після видалення тих, в кого є бал 2 і нижче



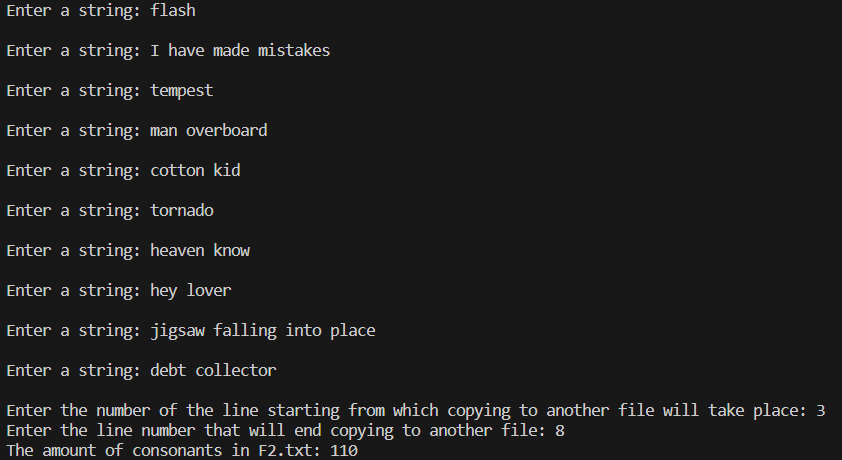
Малюнок 5: Виведення списку школярів пясля додавання нового



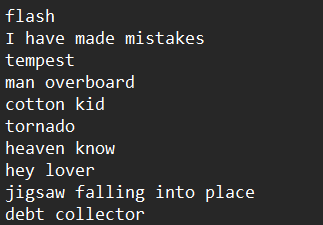
Малюнок 6: Остаточний бінарний файл

Час затрачений на виконання завдання: >20год

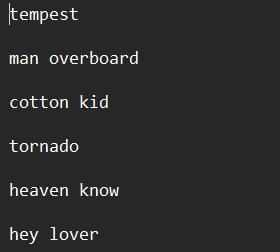
Завдання №3 Лабораторна робота №9 "Ввід-вивід рядків"



Малюнок 7: Успішна компіляція програми із завдання №3



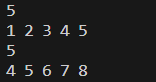
Малюнок 8: Вміст текстового файлу F1



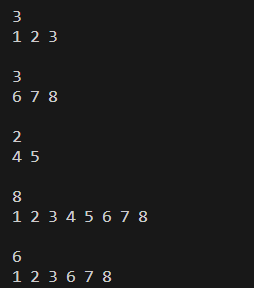
Малюнок 9: Вміст текстового файлу F2

Час затрачений на виконання завдання: 2год

Завдання №4 Лабораторна робота №4 у Algotester



Малюнок 10: Запропонований у Algotester ввід



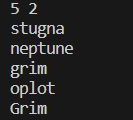
Малюнок 11: Вивід співпадає із прикладом у Algotester, що означає успішну компіляцію



Малюнок 12: Підтвердження правильності коду у Algotester

Час затрачений на виконання завдання: 4год

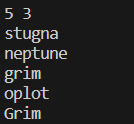
Завдання №5 Лабораторна робота №6 у Algotester



Малюнок 13: Один із запропонованих у Algotester вводів



Малюнок 14: Вивід співпадає із прикладом у Algotester, що означає успішну компіляцію



Малюнок 15: Один із запропонованих у Algotester вводів



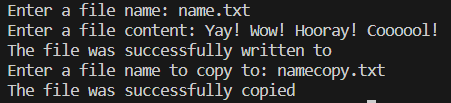
Малюнок 16: Вивід співпадає із прикладом у Algotester, що означає успішну компіляцію



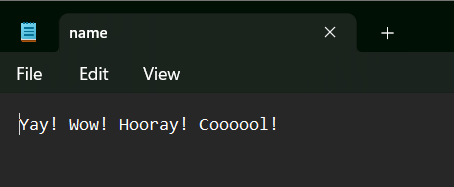
Малюнок 17: Підтвердження правильності коду у Algotester

Час затрачений на виконання завдання: 5год

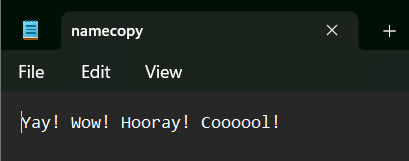
Завдання №6 Практична робота “Робота з текстовими файлами”



Малюнок 18: Успішна компіляція програми із завдання №6



Малюнок 19: Вміст початкового текстового файлу



Малюнок 20: Скопійований із початкового файлу вміст нового

Час затрачений на виконання завдання: 2год

Завдання №7 Власна практична робота “Лабораторна робота №4 у Algotester”



Малюнок 21: Запропонований у Algotester ввід



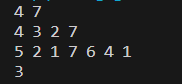
Малюнок 22: Вивід співпадає із прикладом у Algotester, що означає успішну компіляцію



Малюнок 23: Підтвердження правильності коду у Algotester

Час затрачений на виконання завдання: 2год

Завдання №8 Власна практична робота “Хелловін”



Малюнок 24: Перші 3 рядки запропонований у Algotester ввід; останній рядок -- вивід, що співпадає із прикладом у Algotester: успішна компіляція



Малюнок 25: Підтвердження правильності коду у Algotester

Час затрачений на виконання завдання: 20хв

**6. Кооперація з командою:**



Малюнок 26: Скрін із зустрічі з командою у Discord

# **Висновки:**

* Ознайомлений із файлами
* Ознайомлений із системами числення
* Ознайомлений із бінарними файлами
* Ознайомлений із символами та рядковими змінними та текстовими файлами
* Ознайомлений зі стандартною бібліотекою та деталями й методами роботи з нею
* Ознайомлений зі створенням та використанням бібліотек
* Успішне використання нововивченого матеріалу під час виконання лабораторних та практичних робіт