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.»

***Виконала:***

студентка групи ШІ-13

Кузнєцова Анастасія Володимирівна

# **Тема роботи:**

Організація роботи з файлами, систем числення та їхнє використання в програмуванні. Бінарні файли, символи та рядкові змінні, текстові файли. Робота зі стандартною бібліотекою для файлів та створення бібліотек у програмуванні.

# **Мета роботи:**

Вивчення та опрацювання двійкових файлів, організація вводу-виводу структурованої інформації зберіганням на зовнішніх носіях. Також, вивчення символьних і рядкових змінних у мові C++ та робота з текстовими файлами, включаючи ввід-вивід текстової інформації та зберігання на зовнішніх носіях.

# **Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем:

* Тема №1: Потоки вводу і виводу. Базовий файловий ввід і вивід.
* Тема №2: Функціонал push\_back векторів.
* Тема №3: Стандартна бібліотека шаблонів (STL).
* Тема №4: Перерахування (перелічуваний тип даних).
* Тема №5: Структури.

1. Індивідуальний план опрацювання теорії:

* Тема №1: Потоки вводу і виводу. Базовий файловий ввід і вивід.
  + Джерела Інформації:
    - Стаття: https://acode.com.ua/urok-218-potokovi-klasy-i-ryadky/  
      https://acode.com.ua/urok-217-funktsional-klasiv-ostream-i-ios-formatuvannya-vyvodu/  
      https://cplusplus.com/doc/tutorial/files/  
      <https://www.tutorialspoint.com/reading-and-writing-binary-file-in-c-cplusplus>  
      https://acode.com.ua/urok-216-funktsional-klasu-istream/  
      https://acode.com.ua/urok-215-potoky-vvodu-i-vyvodu/  
      <https://www.udacity.com/blog/2021/05/how-to-read-from-a-file-in-cpp.html>
    - http://cpp.dp.ua/potokove-vvedennya-vyvedennya/
  + Що опрацьовано: Ознайомлено та опрацьовано роботу потоків введення-виведення в C++. Опрацьовано можливості здійснення різноманітних операцій вводу-виводу над даними довільних типів та форматів за допомогою відповідних функцій та класів, доступ до яких надається через заголовний файл iostream.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 01.12.23
  + Звершення опрацювання теми: 10.12.23
* Тема №2: Функціонал push\_back векторів
  + Джерела Інформації:
    - Стаття: https://cplusplus.com/reference/vector/vector/push\_back/
  + Що опрацьовано: Як переносити інформацію з файлу у вектор.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 02.12.23
  + Звершення опрацювання теми: 03.12.23
* Тема №3: Стандартна бібліотека шаблонів (STL).
  + Джерела Інформації:
    - Стаття: <https://acode.com.ua/urok-101-algorytmy-v-standartnij-bibliotetsi-s/#toc-5>  
      https://acode.com.ua/urok-207-algorytmy-stl/  
      https://www.geeksforgeeks.org/std-set\_difference-in-cpp/  
      https://www.bestprog.net/uk/2021/09/26/c-standard-template-library-general-concepts-ua/  
      <https://en.cppreference.com/w/cpp/algorithm/set_difference>  
      http://www.kytok.org.ua/post/cplusplus-std-set\_difference-riznytsya-mnozhyn
  + Що опрацьовано: Ознайомлено зі стандартною бібліотекою шаблонів (STL). Опрацьовано, що STL надає готові реалізації поширених структур даних та алгоритмів.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 03.12.23
  + Звершення опрацювання теми: 10.12.23
* Тема №4: Перерахування (перелічуваний тип даних).
  + Джерела Інформації:
    - Стаття: https://acode.com.ua/urok-61-pererahuvannya-perelichuvanyj-typ-danyh/  
      https://acode.com.ua/urok-100-vvedennya-v-iteratory/  
      https://www.simplilearn.com/tutorials/cpp-tutorial/cpp-enum
  + Що опрацьовано: Опрацьовано особливості перелічувального типу даних (enum). Вивчено основний функціонал та особливості використання перерахувань.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 05.12.23
  + Звершення опрацювання теми: 09.12.23
* Тема №5: Структури.
  + Джерела Інформації:
    - Стаття: https://acode.com.ua/urok-64-struktury/
  + Що опрацьовано: Опрацьовано особливості використання структур даних. Як використовувати структури та яка їх зручність.
  + Статус: Ознайомлений
  + Початок опрацювання теми: 01.12.23
  + Звершення опрацювання теми: 10.12.23

# **Виконання роботи:**

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

Завдання №1 **VNS Lab 6**

* Варіант завдання: 8
* Деталі завдання: Виконати ввід рядка, використовуючи функцію gets(s) і перетворити рядок так, щоб всі слова в ньому стали ідентифікаторами. Слова, які складаються тільки із цифр, знищити.
* Важливі деталі для врахування в імплементації програми: Текст містить не більше 255 символів. Слова одне від одного відокремлюються одним або декількома пробілами. Наприкінці тексту ставиться крапка.

Завдання №2 **VNS Lab 8**

* Варіант завдання: 8
* Деталі завдання: Створити двійковий файл із елементами відповідно до структури "Покупець". Вивести вміст файлу. Видалити перші три стрічки та додати 3 у кінець файлу. Після проведених операцій вивести текст файлу.
* Важливі деталі для врахування в імплементації програми: Забезпечити обробку повідомлень про помилки при відкритті файлу та виконанні операцій вводу/виводу.

Завдання №3 **VNS Lab 9**

* Варіант завдання: 8  
  Деталі завдання: Створити текстовий файл F1 та записати в нього інформацію. Виконати завдання: скопіювати з файлу F1 у файл F2 всі рядки, які не містять цифри, і підрахувати кількість рядків у файлі F2, які починаються на букву «А».
* Важливі деталі для врахування в імплементації програми: Файл F1 складається не менше ніж з 10 рядків.

Завдання №4 **Algotester Lab 4**

* Варіант завдання: 1
* Деталі завдання: Вводиться два цілих числових масиви розміром N та M. Визначити різницю N-M, визначити різницю M-N, знайти перетин обох масивів, знайти об'єднання обох масивів та знайти симетричну різницю обох масивів.
* Важливі деталі для врахування в імплементації програми: Використати стандартну бібліотеку шаблонів.

Завдання №5: **Algotester Lab 6**

* Варіант завдання: 1
* Деталі завдання: Задається N слів та число K. Знайти слово яке повторюється K разів. Порахувати кількість літер у цьому слові та вивести її на екран. Вивести малі літери цього слова у зворотному алфавітному порядку через пробіл. У випадку, якщо таких букв немає, необхідно вивести "Empty!". Числом M позначається кількість унікальних букв на виході. У другому рядку представляються унікальні букви через пробіли.
* Важливі деталі для врахування в імплементації програми: Великі та маленькі букви вважаються однаковими. Кожну букву слід виводити лише один раз.

Завдання №6: **Class Practice Work**

Task 1:

* Деталі завдання: Створити файл із заданим ім'ям; у випадку існуючого файлу, перезаписати його вміст. Провести запис в файл вмісту рядка content та здійснити зчитування вмісту зі стандартного вводу. Повернути статус операції: "Success" у випадку успішного виконання усіх операцій, "Failure" у випадку невдалого створення файлу або збою під час операцій відкриття.
* Важливі деталі для врахування в імплементації програми: Використовувати перелічувальний тип даних.

Task 2:

* Деталі завдання: Копіювати вміст файла з ім’ям file\_from у файл з ім’ям file\_to. Повернути статус операції: "Success" у випадку успішного виконання усіх операцій, "Failure" у випадку невдалого створення файлу або збою під час операцій відкриття.
* Важливі деталі для врахування в імплементації програми: Використовувати перелічувальний тип даних. Імена file\_from та file\_to можуть бути вказані як повний, так і відносний шлях.

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

Програма №1 **VNS Lab 6**

* Планований час на реалізацію: 5 годин
* Важливі деталі для врахування в імплементації: Текст містить не більше 255 символів. Слова одне від одного відокремлюються одним або декількома пробілами. Наприкінці тексту ставиться крапка.

Програма №2 **VNS Lab 8**

* Планований час на реалізацію: 1 день
* Важливі деталі для врахування в імплементації: Забезпечити обробку повідомлень про помилки при відкритті файлу та виконанні операцій вводу/виводу.

Програма №3 **VNS Lab 9**

* Планований час на реалізацію: 9 годин
* Важливі деталі для врахування в імплементації: Файл F1 складається не менше ніж з 10 рядків.

Програма №4 **Algotester Lab 4**

* Планований час на реалізацію: 4 години
* Важливі деталі для врахування в імплементації: Використати стандартну бібліотеку шаблонів.

Програма №5 **Algotester Lab 6**

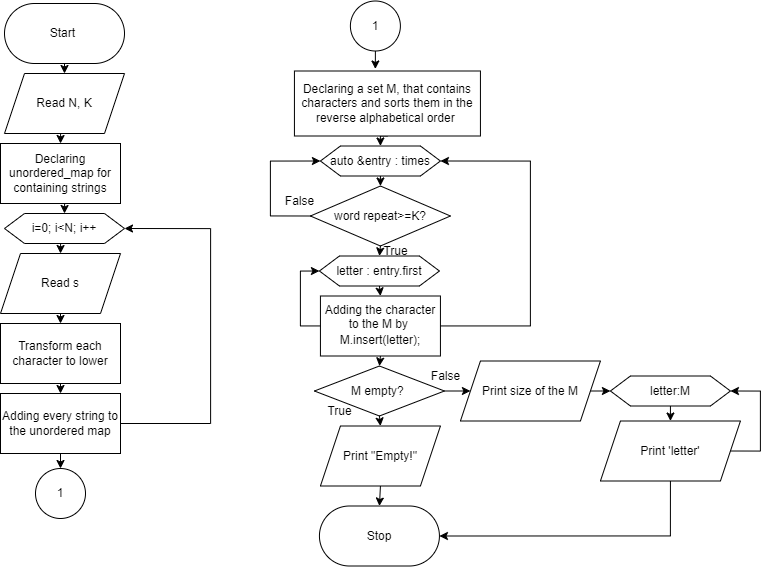


Рисунок Діаграма до Algotester 6v1

* Планований час на реалізацію: 8 годин
* Важливі деталі для врахування в імплементації: Великі та маленькі букви вважаються однаковими. Кожну букву слід виводити лише один раз.

Програма №6 **Class Practice Work**

* Планований час на реалізацію: 4 години
* Важливі деталі для врахування в імплементації: Використовувати перелічувальний тип даних.

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

Файл settings.json в якому зберігаються налаштування VS Code

{

*"workbench.colorTheme"*: "Monokai",

*"code-runner.runInTerminal"*: true,

*"C\_Cpp.default.compilerPath"*: "d:\\C C++\\test.c",

*"cmake.configureOnOpen"*: true,

*"terminal.integrated.defaultProfile.windows"*: "Windows PowerShell",

*"terminal.integrated.profiles.windows"*: {

*"PowerShell"*: {

*"source"*: "PowerShell",

*"icon"*: "terminal-powershell"

        },

*"Command Prompt"*: {

*"path"*: [

                "${env:windir}\\Sysnative\\cmd.exe",

                "${env:windir}\\System32\\cmd.exe"

            ],

*"args"*: [],

*"icon"*: "terminal-cmd"

        },

*"Git Bash"*: {

*"source"*: "Git Bash"

        },

*"bash (MSYS2)"*: {

*"path"*: "C:\\msys64\\usr\\bin\\bash.exe",

*"args"*: [

                "--login",

                "-i"

            ]

        },

*"Windows PowerShell"*: {

*"path"*: "C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe"

        }

    },

*"workbench.colorCustomizations"*: {}

}

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

Завдання №1 **VNS Lab 6**

#include <iostream>

#include <cstring>

using *namespace* std;

*void* onlynumbers(*char* *str*[])

{

*char* \*token;

    token=strtok(*str*, " ");

    while (token != NULL)

    {

*bool* digits=true;

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

        {

            if (!isdigit(token[i]))

            {

                digits=false;

                break;

            }

        }

        if (!digits)

        {

            cout<<token<<".";

        }

    token = strtok(NULL, " ");

    }

}

*int* main()

{

*char* str[256];

    cout<<"Enter a string\n";

    gets(str);

    onlynumbers(str);

return 0;

}

Програма виконує введення рядка за допомогою функції gets(s) та перетворює його. Всі рядки, що складають тільки з цифр знищуються. Так, як текст може містити не більше 255 символів, то записується масив char з розміром 256, бо треба передбачати позицію нульового символу.

Pull request: https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/783/files#diff-307d6d9cde140a1141078ee2a367291b36a08e1b05d7cb2289f68b6192577162

Завдання №2 **VNS Lab 8**

#include <iostream>

#include <fstream>

#include <string>

#include <vector>

using *namespace* std;

*struct* Customer

{

*char* name[40];

*char* surname[40];

*char* patronymic[40];

*char* address[40];

*char* number[100];

*char* card\_number[40];

};

*int* main()

{

    Customer m;

    ofstream info("Customers\_information.txt");

    if(!info)

    {

        cerr << "Customers.txt could not be opened";

        exit(1);

    }

    cout << "Write a name: ";  cin >> m.name;  info  << m.name << "\n";

    cout << "Write a surname: ";  cin >> m.surname;  info << m.surname << "\n";

    cout << "Write a patronymic: ";  cin >> m.patronymic;  info << m.patronymic << "\n";

    cout << "Write an address: ";  cin >> m.address;  info << m.address << "\n";

    cout << "Write a number: ";  cin >> m.number;  info << m.number << "\n";

    cout << "Write a card\_number: ";  cin >> m.card\_number;  info << m.card\_number << "\n";

    info.close();

    ifstream printinfo("Customers\_information.txt");

    if (!printinfo)

    {

        cerr << "Customers\_information.txt could not be opened";

        exit(2);

    }

    string print;

    while (getline(printinfo, print))

    {

        cout << print << "\n";

    }

    printinfo.close();

    vector<Customer> people;

    ifstream infile("Customers\_information.txt");

    if (!infile)

    {

        cerr << "Customers\_information.txt could not be opened";

        exit(3);

    }

    while (infile >> m.name >> m.surname >> m.patronymic >> m.address >> m.number >> m.card\_number)

    {

        people.push\_back(m);

    }

    infile.close();

    people.erase(people.begin(), people.begin()+3);

    Customer newCustomer;

    cout << "Write a name: "; cin >> newCustomer.name; people.push\_back(newCustomer);

    cout << "Write a surname: "; cin >> newCustomer.surname; people.push\_back(newCustomer);

    cout << "Write a patronymic: "; cin >> newCustomer.patronymic; people.push\_back(newCustomer);

    ofstream tovector("Customers\_information.txt");

    if (!tovector)

    {

        cerr << "Customers\_information.txt could not be opened";

        exit(4);

    }

    for (const *auto*& person : people)

    {

        tovector << person.address << "\n";

        tovector << person.number << "\n";

        tovector << person.card\_number << "\n";

        tovector << person.name << "\n";

        tovector << person.surname << "\n";

        tovector << person.patronymic << "\n";

    }

    tovector.close();

    ifstream modyfiedinfo("Customers\_information.txt");

    if (!modyfiedinfo)

    {

        cerr << "Customers\_information.txt could not be opened";

        exit(6);

    }

    string line;

    while (getline(modyfiedinfo, line))

    {

        cout << line << endl;

    }

    modyfiedinfo.close();

    return 0;

}

Програма виконує створення двійкового файлу з елементами структури "Покупець". Після цього вона виводить вміст файлу. Далі видаляє перші три стрічки та додає три нові у кінець файлу. Після проведених операцій програма виводить змінений текст файлу. Якщо файл з якихось причин не відкривається, виводить помилку через потік помилок cerr та виходить з виконання програми.

Допоміжні джерела:   
<https://stackoverflow.com/questions/9216354/reading-and-printing-an-entire-file-in-binary-mode-using-c>  
<https://stackoverflow.com/questions/6406356/how-to-write-vector-values-to-a-file>

Pull request: https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/783/files#diff-90c56a15e39017c927ab1ebd974ac0f3a77a0fbc4477f56f1b275843358079a2

Завдання №3 **VNS Lab 9**

#include <iostream>

#include <fstream>

#include <string>

using *namespace* std;

*bool* digit(const string& *line*) {

    for (*char* num : *line*) {

        if (isdigit(num)) {

            return true;

        }

    }

    return false;

}

*int* main()

{

    ofstream firstfile("F1.txt");

    if (!firstfile)

    {

        cerr<<"Unable to open the F1.txt"<<"\n";

        exit(1);

    }

    firstfile<<"Anastasia\n";

    firstfile<<"Kuznietsova\n";

    firstfile<<"Volodumurivna\n";

    firstfile<<"03032006\n";

    firstfile<<"17\_years\_old\n";

    firstfile<<"Like\_cats\n";

    firstfile<<"Freshman\n";

    firstfile<<"AI-13\n";

    firstfile<<"Lviv\_Politechnic\n";

    firstfile<<"I\_have\_Yoda\n";

    firstfile.close();

    ifstream printfirst("F1.txt");

    string words;

    if (!printfirst)

    {

        cerr<<"Unable to open the F1.txt"<<"\n";

        exit(2);

    }

    while (printfirst>>words)

    {

        cout<<words<<"\n";

    }

    cout<<"\n";

    printfirst.close();

    ifstream openfile("F1.txt");

    if (!openfile)

    {

        cerr<<"Unable to open the F1.txt"<<"\n";

        exit(3);

    }

    ofstream secondfile("F2.txt");

    if (!secondfile)

    {

        cerr << "Unable to open F2.txt" << endl;

        exit(4);

    }

    string line;

    while (getline(openfile, line))

    {

        if(!digit(line))

        {

            secondfile<<line<<"\n";

        }

    }

    openfile.close();

    secondfile.close();

    ifstream printsecond("F2.txt");

    string end;

*int* A = 0;

    if (!printsecond)

        {

            cerr << "Unable to open F2.txt" << endl;

            exit(5);

        }

    while (printsecond>>line)

        {

            cout<<line<<"\n";

        if (line[0]=='A')

            {

                A++;

            }

        }

    cout<<"\n";

    printsecond.close();

    cout<<A<<"\n";

    return 0;

}

Програма створює текстовий файл F1 та записує в нього інформацію. Після цього вона копіює з файлу F1 у файл F2 всі рядки, які не містять цифри. Після цього, програма рахує та виводить кількість рядків у файлі F2, які починаються на букву «А». При помилці програма виводить відповідний текст та закінчує роботу.

Pull request: https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/783/files#diff-abe74eedc3854484b453861c2cb0235c06619432f24e7799080ea7fe2126f706

Завдання №4 **Algotester Lab 4**

#include <iostream>

#include <algorithm>

#include <vector>

#include <string>

using *namespace* std;

*int* A, B;

*int* main()

{

*int* A, B;

    cin>>A;

*int* N[A];

    for(*int* i=0; i<A; i++)

    {

        cin>>N[i];

    }

    cin>>B;

*int* M[B];

    for(*int* i=0; i<B; i++)

    {

        cin>>M[i];

    }

    sort(N, N+A);

    sort(M, M+B);

    const *int* maxsize = A + B;

*int* v[maxsize];

*int*\* a = set\_difference(N, N+A, M, M+B, v);

    cout << a-v <<"\n";

    for (*int*\* i=v; i!=a; i++)

    {

        cout<<\*i<<" ";

    }

    cout<<"\n";

*int*\* b=set\_difference(M, M+B, N, N+A, v);

    cout << b - v << "\n";

    for (*int*\* i=v; i!=b; i++)

    {

        cout<<\*i<<" ";

    }

    cout<<"\n";

*int*\* c=set\_intersection(M, M+B, N, N+A, v);

    cout<<c-v<<"\n";

    for (*int*\* i=v; i!=c; i++)

    {

        cout<<\*i<<" ";

    }

    cout<<"\n";

*int*\* d=set\_union(M, M+B, N, N+A, v);

    cout<<d-v<<"\n";

    for (*int*\* i=v; i!=d; i++)

    {

        cout<<\*i<<" ";

    }

    cout<<"\n";

*int*\* e=set\_symmetric\_difference(M, M+B, N, N+A, v);

    cout<<e-v <<"\n";

    for (*int*\* i=v; i!=e; i++)

    {

        cout<<\*i<<" ";

    }

return 0;

}

Програма запрошує ввести два цілі числа – розміри масивів N та M. Потім відповідно введення елементів цих масивів, за допомогою циклу. Після цього вона визначає різницю N-M та різницю M-N, знаходить перетин обох масивів, знаходить об'єднання обох масивів та знаходить симетричну різницю обох масивів, використовуючи стандартну бібліотеку шаблонів.

Допоміжні джерела:  
https://stackoverflow.com/questions/56138478/i-dont-understand-how-to-use-set-difference-correctly  
Pull request: https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/783/files#diff-1ba245754e8b09cd81fadf778767e2149a8a2042d8381d169730677c8316d9a5

Завдання №5 **Algotester Lab 6**

#include <iostream>

#include <algorithm>

#include <vector>

#include <set>

#include <unordered\_map>

using *namespace* std;

*int* main()

{

*int* N, K;

    string s;

    cin>>N>>K;

    unordered\_map<string, *int*> times;

    for (*int* i=0; i<N; i++)

    {

        cin>>s;

        transform(s.begin(), s.end(), s.begin(), ::tolower);

        times[s]++;

    }

    set <*char*, greater<*char*>> M;

    for (const *auto* &entry : times)

    {

        if (entry.second>=K)

        {

            for (*char* letter : entry.first)

            {

                M.insert(letter);

            }

        }

    }

    if (M.empty())

    {

        cout<<"Empty!";

        return 0;

    }

    else

    {

        cout<<M.size()<<"\n";

        for (*char* letter:M)

        {

        cout<<letter<<" ";

        }

    }

return 0;

}

Спочатку вводяться N – максимальна кількість слів та число K – скільки разів слово повторюється, з яким треба виконувати далі дії. Далі програма виводить кількість літер, M, у цьому слові. Всі великі літери перетворюються на малі. У відповідь програма виводить літери цього слова у зворотному алфавітному порядку через пробіл. У випадку, якщо таких букв немає, виводить "Empty!".   
Допоміжні джерела: <https://stackoverflow.com/questions/40667989/find-length-of-the-stringexcluding-spaces>https://www.geeksforgeeks.org/find-repeated-character-present-first-string/  
https://prepinsta.com/cpp-program/to-find-duplicate-characters-in-a-string/  
https://www.quora.com/How-do-I-print-the-content-of-a-file-using-an-fstream-in-C

Pull request: https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/783/files#diff-f37f39d95037685c0e850c066e251c25386ff2be2774c9243ae74cf9de1f95ff

Завдання №6: **Class Practice Work**

**Task 1**

#include <iostream>

#include <fstream>

using *namespace* std;

*enum* FileOpResult {Success, Failure};

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

{

    if (!*name*||!*content*)

    {

        cerr<<"Something's gone wrong";

        return FileOpResult::Failure;

    }

    ofstream file(*name*);

    if (!file)

    {

        cerr<<"File can't be opened";

        return FileOpResult::Failure;

    }

    file << *content*;

    if (!file)

    {

        cerr<<"File can't be opened";

        return FileOpResult::Failure;

    }

    file.close();

    return FileOpResult::Success;

}

*int* main()

{

    const *char* \*name = "File.txt";

    const *char* \*content= "I want to sleep";

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

    if (result == FileOpResult::Success)

    {

        cout<<"Success";

    }

    else

    {

        cout<<"Failure";

    }

    return 0;

}

У програмі використовується оператор перерахування для використання повернення Success чи Failure. Створюється файл з відповідною назвою. Програма записує у нього вміст рядка content, “I want to sleep”, та зчитує його. Після цього повертає "Success" якщо все виконалось успішно та "Failure" – якщо щось не виконалось, або відбувся збій.

Pull request: https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/783/files#diff-774d11f592984cddc0950fa3e20a915ed17c43f328afe1add2a4b0e5e9a55aec

**Task 2**

#include <iostream>

#include <fstream>

#include <string>

using *namespace* std;

*enum* FileOpResult {Success, Failure};

FileOpResult copy\_file(string *file\_from*, string *file\_to*)

{

    ifstream from(*file\_from*);

    if (!from)

    {

        cerr<<"File can't be opened";

        return FileOpResult::Failure;

    }

    ofstream to(*file\_to*);

    if (!to)

    {

        cerr<<"File can't be opened";

        return FileOpResult::Failure;

    }

    string line;

    while (getline(from, line))

    {

        to<<line<<"\n";

    }

    from.close();

    to.close();

    return FileOpResult::Success;

}

*int* main()

{

    string file\_from = "D:/paradigms/Epic 5/F1.txt";

    string file\_to = "D:/paradigms/Epic 5/F2.txt";

    ofstream data (file\_from);

    if (!data)

    {

        cerr<<"File can't be opened";

        return FileOpResult::Failure;

    }

    data<<"MEOW MEOW MEOW MEOW MEOW MEOW MEOW MEOW MEOW MEOW MEOW MEOW MEOW";

    data.close();

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

    if (result == FileOpResult::Success)

    {

        cout<<"Success";

    }

    else

    {

        cerr<<"Failure";

    }

    return 0;

}

Використовується перелічувальний тип даних, для подальшого виведення результату. Програма копіює вміст файла file\_from у файл file\_to. Після виконання операцій повертає статус операції: "Success" якщо все виконалось успішного, або "Failure" – якщо не успішно.

Допоміжні джерела:  
https://www.geeksforgeeks.org/cpp-program-to-read-content-from-one-file-and-write-it-into-another-file/  
Pull request: https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/783/files#diff-4629bacef6e2d07b67f6846aedb97801ce04d598972944b95f0d402c18783da1

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

Завдання №1 **VNS Lab 6**

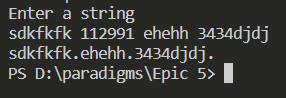


Рисунок Виконання VNS lab 6

Час, затрачений на виконання завдання: 4 години

Завдання №2 **VNS Lab 9**

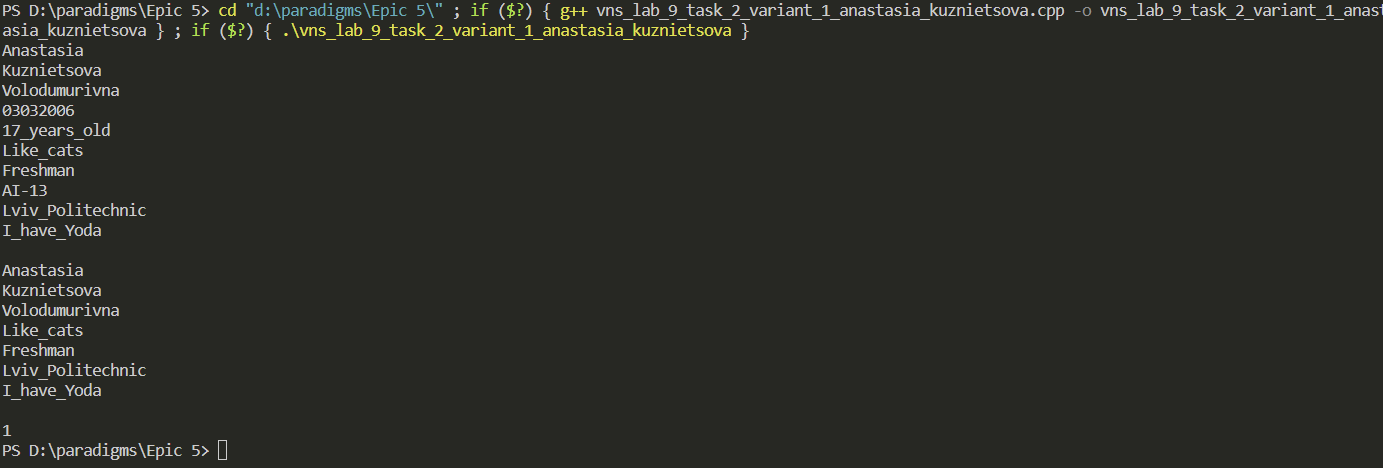


Рисунок Виконання VNS lab 9

Час, затрачений на виконання завдання: 6 годин

Завдання №3 **Algotester Lab 4**

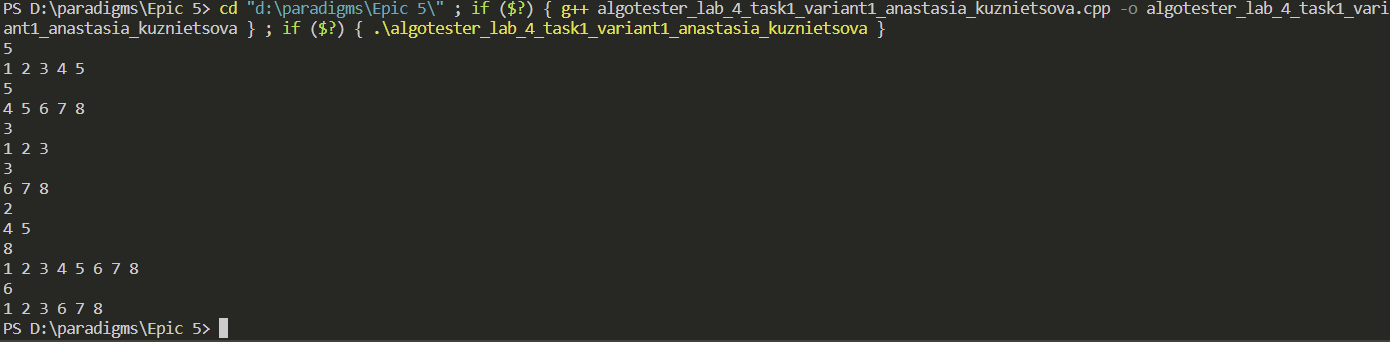


Рисунок Виконання Algotester lab 4v1

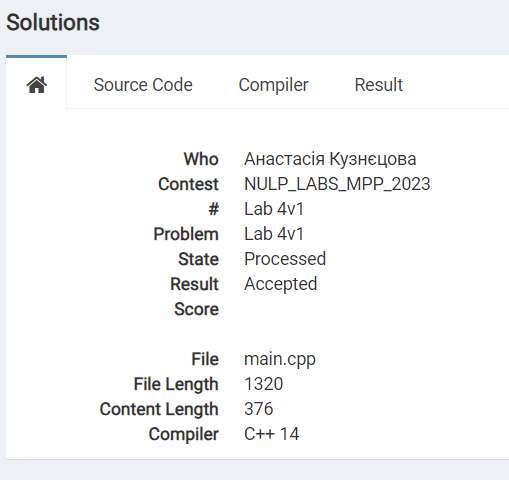


Рисунок Скріншот Algotester lab 4v1

Час, затрачений на виконання завдання: 8 годин

Завдання №4 **Algotester Lab 6**

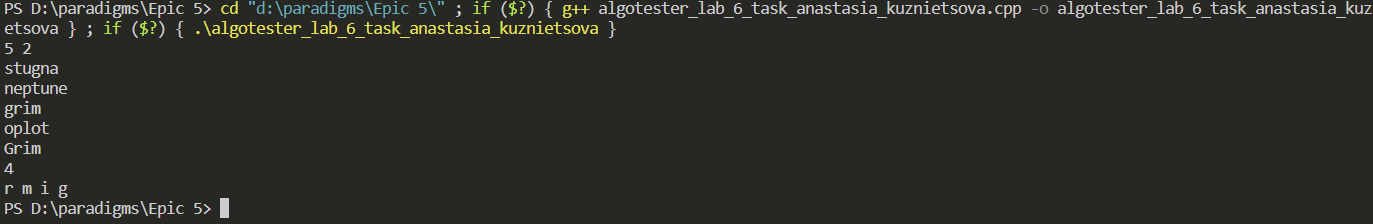


Рисунок Виконання Algotester lab 6v1

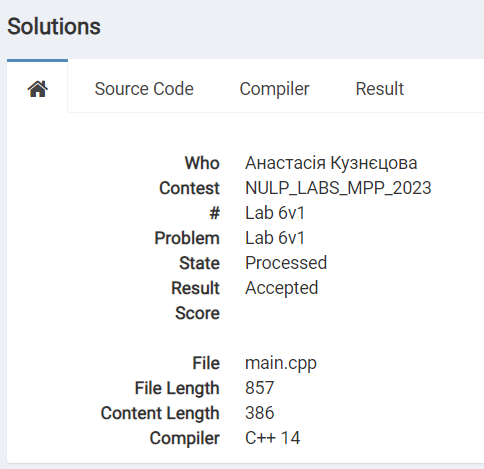


Рисунок Скріншот Algotester lab 6v1

Час, затрачений на виконання завдання: 2 дні

Завдання №5 **Class Practice Work**

**Task 1**

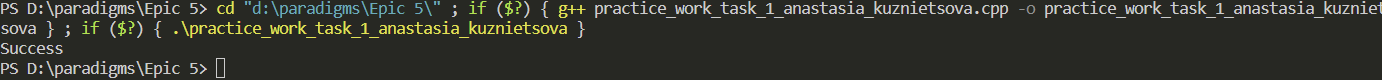
****

Рисунок Виконання task 1 class practice work

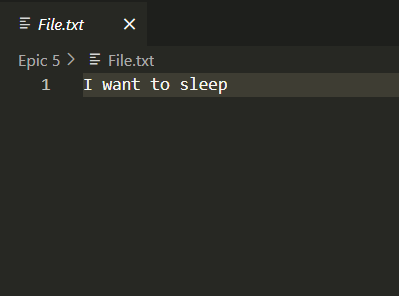


Рисунок Створений файл task 1 class practice work

Програма вивела “Success” бо створення файлу та введення в нього даних пройшло успішно.

**Task 2**



Рисунок Виконання task 2 class practice work

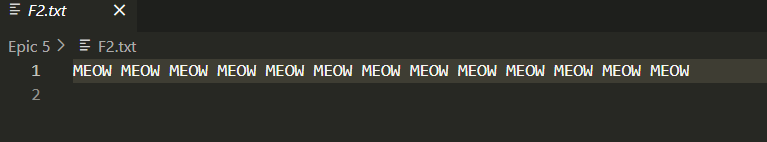


Рисунок Створений файл task 2 class practice work

Програма вивела “Success” бо створення двох файлів та перенесення стрічки з одного у інший пройшло успішно.

Час, затрачений на виконання завдання: 5 годин.

# **Pull-request:** https://github.com/artificial-intelligence-department/ai\_programming\_playground/pull/783

# **Висновки:**

Ознайомлено та опрацьовано основи роботи з файлами. Розглянуто принципи організації двійкових та текстових файлів, використання систем числення при зберіганні інформації. Також вивчено роботу зі стандартними бібліотеками вводу-виводу, механізм створення власних бібліотек.