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. Бінарні Файли. Символи і Рядкові Змінні та Текстові Файли. Стандартна бібліотека та деталі/методи роботи з файлами. Створення й використання бібліотек.» ***з дисципліни:*** «Основи програмування»

до:

ВНС Лабораторної Роботи № 6

ВНС Лабораторної Роботи № 8

ВНС Лабораторної Роботи № 9

Алготестер Лабораторної Роботи №4

Алготестер Лабораторної Роботи №6

Практичних Робіт до блоку №5

Виконав:

Студент групи ШІ-12

Горішний Микола

Львів – 2024

Тема роботи: Файли. Бінарні Файли. Символи і Рядкові Змінні та Текстові Файли. Стандартна бібліотека та деталі/методи роботи з файлами. Створення й використання бібліотек.

Мета роботи:

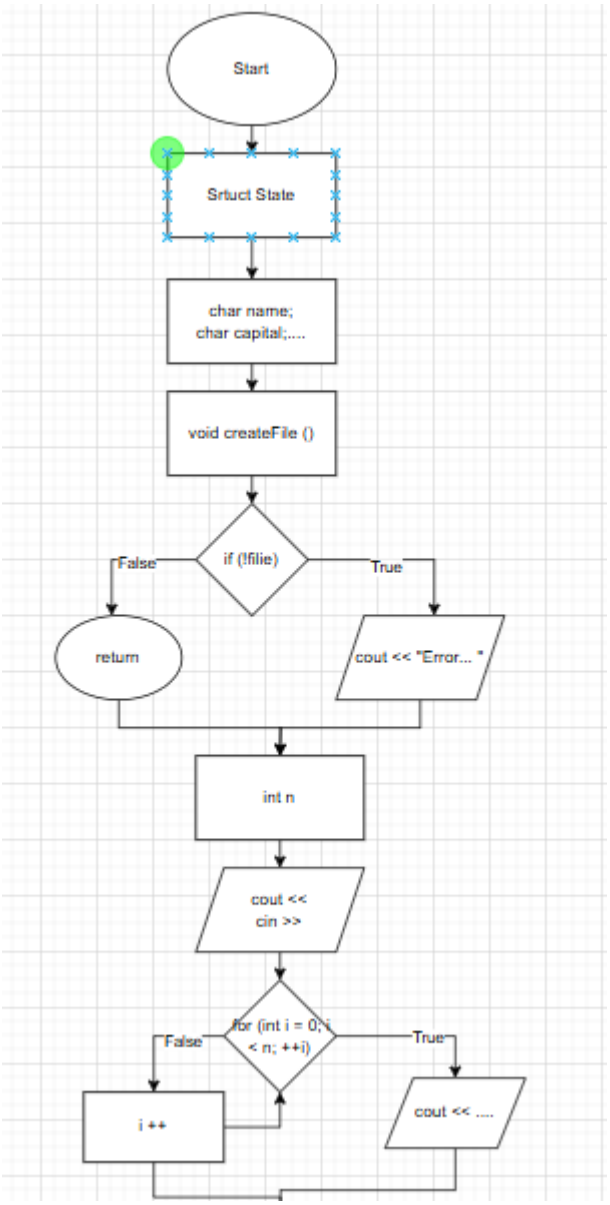
Дослідити основи роботи з файлами у мові програмування C++, зокрема розглянути та освоїти принципи роботи з текстовими та бінарними файлами. Опанувати операції введення та виведення символів і рядкових змінних у файл, а також ознайомитися зі стандартною бібліотекою C++ для роботи з файлами. Навчитися створювати власні бібліотеки та використовувати їх у проектах, організовуючи код для повторного використання та покращення його структури.

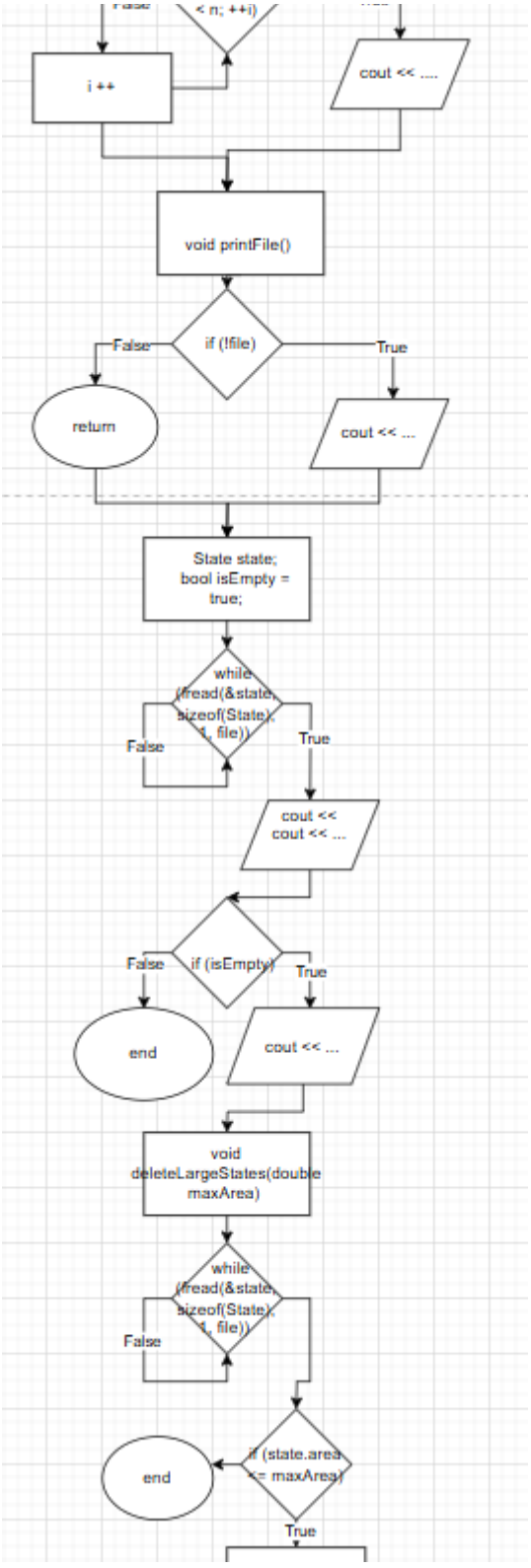
Джерала інформації:

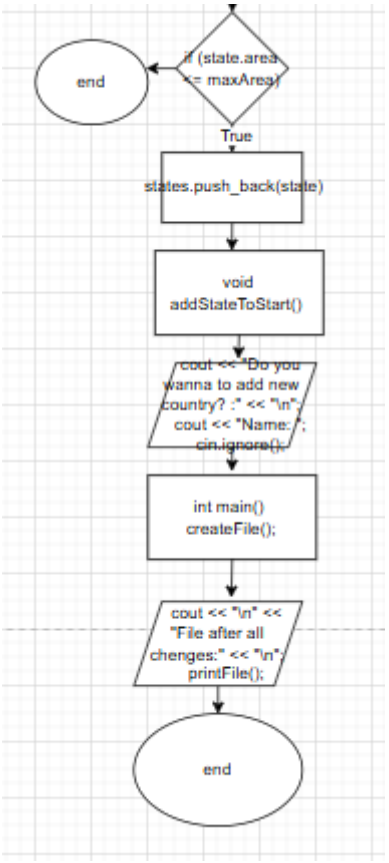
--https://www.youtube.com/watch?v=2lzVB8bkM8o;
--https://www.youtube.com/watch?v=FeNqHytI0fA;
--https://acode.com.ua/uroki-po-cpp/;

Виконання роботи

Requirements management (understand tasks) and design activities draw flow diagram


Start
Srtuct State
char name;
char capital;....
void createFile ()
if (!filie)
False
True
return
cout << "Error... "
int n
cout <<
cin >>
for (int i = 0; i
< n; ++i)
False
True
i ++
cout << ....


< n; ++i)
i ++
cout << ....
void printFile()
if (!file)
False
True
return
cout << ...
State state;
bool isEmpty =
true;
while
(fread(&state,
sizeof(State),
1, file))
False
True
cout <<
cout << ...
if (isEmpty)
False
True
end
cout << ...
void
deleteLargeStates(double
maxArea)
while
(fread(&state,
sizeof(State),
1, file))
False
end
if (state.area
<= maxArea)
True


end
if (state.area
<= maxArea)
True
states.push_back(state)
void
addStateToStart()
cout << "Do you
wanna to add new
country? :" << "\n";
cout << "Name: ";
cin.ignore();
int main()
createFile();
cout << "\n" <<
"File after all
chenges:" << "\n";
printFile();
end

Lab# programming: VNS_Lab_6_Task_1 (3) (30 хв)

```cpp
#include <iostream>
#include <cstring>
#include <cctype>
using namespace std;

int main()
{
    char s[256];
    cout << "Enter elements: ";
    fgets(s, sizeof(s), stdin);

    s[strcspn(s, "\n")] = 0;

    char* word = strtok(s, " ");

    cout << "Elements with 1 figure :" << endl;
    while (word != nullptr)
    {
        int digitCount = 0;

        for (int i = 0; word[i] != '\0'; i++)
        {
            if (isdigit(word[i]))
            {
                digitCount++;
            }
        }

        if (digitCount == 1)
        {
            cout << word << endl;
        }

        word = strtok(nullptr, " ");
    }

    return 0;
}
```

```
Enter elements: hello 12 my beautiful 1 world 2 and friends 23
Elements with 1 figure :
1
2
```

Lab# programming: VNS_Lab_8_Task_1 (3) (1 год 20 хв)

2. Постановка 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 "Держава":

- назва;
- столиця;
- чисельність населення;
- займана площа.

```cpp
#include <iostream>
#include <cstdio>
#include <vector>
#include <cstring>
using namespace std;

struct State
{
    char name[100];
    char capital[50];
    int population;
    double area;
};

const char* filename = "states.bin";

void createFile()
{
    FILE* file = fopen(filename, "wb");
    if (!file)
    {
        cerr << "Error with file." << "\n";
        return;
    }

    int n;
    cout << "Enter number of country: ";
    cin >> n;

    for (int i = 0; i < n; ++i)
    {
        State state;
        cout << "Name: ";
        cin.ignore();
        cin.getline(state.name, 100);
        cout << "Capital: ";
        cin.getline(state.capital, 50);
        cout << "Population: ";
        cin >> state.population;
        cout << "Square: ";
```

```
}

void deleteLargeStates(double maxArea)
{
    FILE* file = fopen(filename, "rb");
    if (!file) {
        cerr << "Error with file for reading" << "\n";
        return;
    }

    vector<State> states;
    State state;

    while (fread(&state, sizeof(State), 1, file))
    {
        if (state.area <= maxArea) {
            states.push_back(state);
        }
    }

    fclose(file);

    file = fopen(filename, "wb");
    if (!file)
    {
        cerr << "Error with file for looking" << "\n";
        return;
    }

    for (const auto& s : states)
    {
        fwrite(&s, sizeof(State), 1, file);
    }

    fclose(file);
}
```

```
        cout << "Square: ";
        cin >> state.area;

        fwrite(&state, sizeof(State), 1, file);
    }

    fclose(file);
}

void printFile()
{
    FILE* file = fopen(filename, "rb");
    if (!file)
    {
        cerr << "Error with file for reading" << "\n";
        return;
    }

    State state;
    bool isEmpty = true;

    while (fread(&state, sizeof(State), 1, file))
    {
        isEmpty = false;
        cout << "Name: " << state.name
            << ", Capital: " << state.capital
            << ", Population: " << state.population
            << ", Square: " << state.area << " км²" << "\n";
    }

    if (isEmpty)
    {
        cout << "File is empty" << "\n";
    }

    fclose(file);
}
```

```
    fclose(file);

    file = fopen(filename, "wb");
    if (!file)
    {
        cerr << "Error with file for looking" << "\n";
        return;
    }

    for (const auto& s : states)
    {
        fwrite(&s, sizeof(State), 1, file);
    }

    fclose(file);
}

void addStateToStart()
{
    State newState;
    cout << "Do you wanna to add new country? :" << "\n";
    cout << "Name: ";
    cin.ignore();
    cin.getline(newState.name, 100);
    cout << "Capital: ";
    cin.getline(newState.capital, 50);
    cout << "Population : ";
    cin >> newState.population;
    cout << "Square: ";
    cin >> newState.area;

    FILE* file = fopen(filename, "rb");
    if (!file)
    {
        cerr << "Error with file for reading" << "\n";
        return;
    }
```

```
        return;
    }

    vector<State> states;
    State state;

    while (fread(&state, sizeof(State), 1, file))
    {
        states.push_back(state);
    }

    fclose(file);

    file = fopen(filename, "wb");
    if (!file) {
        cerr << "Error with file for writing" << "\n";
        return;
    }

    fwrite(&newState, sizeof(State), 1, file);

    for (const auto& s : states) {
        fwrite(&s, sizeof(State), 1, file);
    }

    fclose(file);
}

int main()
{
    createFile();

    cout << "\n" << "File after all chenges:" << "\n";
    printFile();

    double maxArea;
    cout << "\n" << "Max square available: ";
    cin >> maxArea;
    deleteLargeStates(maxArea);
```

```
    cin >> maxArea;
    deleteLargeStates(maxArea);

    cout << "\n" << "List contest after deleting too heavy country:" << "\n";
    printFile();

    addStateToStart();

    cout << "\n" << "List contest after adding new country:" << "\n";
    printFile();

    return 0;
}
```

```
Enter number of country: 3
Name: Ukraine
Capital: Kiev
Population: 300
Square: 600
Name: USA
Capital: Washington
Population: 800
Square: 1000
Name: Poland
Capital: Warshawa
Population: 400
Square: 500

File after all chenges:
Name: Ukraine, Capital: Kiev, Population: 300, Square: 600 км²
Name: USA, Capital: Washington, Population: 800, Square: 1000 км²
Name: Poland, Capital: Warshawa, Population: 400, Square: 500 км²

Max square available: 800

List contest after deleting too heavy country:
Name: Ukraine, Capital: Kiev, Population: 300, Square: 600 км²
Name: Poland, Capital: Warshawa, Population: 400, Square: 500 км²
Do you wanna to add new country? :
```

Task 5 - Lab# programming: VNS_Lab_9_Task_1 (3) (1 год)

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

Виконати завдання.

3.

1) Скопіювати у файл F2 тільки ті рядки з F1, які починаються й закінчуються на ту саму букву.
2) Підрахувати кількість символів в F2.

```cpp
#include <iostream>
#include <fstream>
#include <sstream>
#include <string>
using namespace std;

int main()
{
    ofstream fileF1("F1.txt");
    if (!fileF1) {
        cerr << "Error for opening file F1.txt " << endl;
        return 1;
    }


    fileF1 << "Bob" << endl;
    fileF1 << "Mike " << endl;
    fileF1 << "Maus" << endl;
    fileF1 << "Rose " << endl;
    fileF1 << "Diana " << endl;
    fileF1 << "Viki " << endl;
    fileF1 << "Ben " << endl;
    fileF1 << "Michael " << endl;
    fileF1 << "Stensy " << endl;
    fileF1 << "Amelia " << endl;
    fileF1.close();


    ifstream fileF1Read("F1.txt");
    ofstream fileF2("F2.txt");
    if (!fileF1Read || !fileF2)
    {
        cerr << "Error for opening file" << endl;
        return 1;
    }

    string line;
    while (getline(fileF1Read, line))
    {

```

```
    string line;
    while (getline(fileF1Read, line))
    {

        if (!line.empty() && line[0] == 'B')
        {
            fileF2 << line << endl;
        }
    }
    fileF1Read.close();
    fileF2.close();


    ifstream fileF2Read("F2.txt");
    int wordCount = 0;
    string word;
    while (fileF2Read >> word)
    {
        wordCount++;
    }
    fileF2Read.close();

    cout << "Amount of similar words in  F2: " << wordCount << endl;

    return 0;
}
```

Lab# programming: Algotester_Lab_4 (2) (1 год 10 хв)

```cpp
#include <iostream>
#include <vector>
#include <set>
#include <algorithm>
using namespace std;

int main()
{
    int N, K;
    cin >> N >> K;

    vector<int> a(N);
    for (int i = 0; i < N; ++i)
    {
        cin >> a[i];
    }

    set<int> unique_elements(a.begin(), a.end());
    vector<int> unique_array(unique_elements.begin(), unique_elements.end());

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

    K %= unique_array.size();

    rotate(unique_array.begin(), unique_array.begin() + K, unique_array.end());


    cout << unique_array.size() << endl;
    for (int num : unique_array)
    {
        cout << num << " ";
    }
    cout << endl;


    return 0;
}
```

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	Дії
14 годин тому	C++ 20	Зараховано	0.003	1.195	Перегляд

Lab# programming: Algotester_Lab_4 (1) (1 год 10 хв)

```cpp
#include <iostream>
#include <algorithm>

using namespace std;

void results(const int* result, int size)
{
    cout << size << "\n";
    for (int i = 0; i < size; i++)
    {
        cout << result[i] << " ";
    }
    cout << "\n";
}

void difference(const int* n, int N, const int* m, int M)
{
    int result[1000], resultSize = 0;

    int i = 0, j = 0;
    while (i < N && j < M)
    {
        if (n[i] < m[j])
        {
            result[resultSize++] = n[i++];
        }
        else if (n[i] > m[j])
        {
            j++;
        }
        else
        {
            i++;
            j++;
        }
    }

    while (i < N)
    {
        result[resultSize++] = n[i++];
```

```
    {
        result[resultSize++] = n[i++];
    }

    results(result, resultSize);
}

void section(const int* n, int N, const int* m, int M)
{
    int result[1000], resultSize = 0;

    int i = 0, j = 0;
    while (i < N && j < M)
    {
        if (n[i] < m[j])
        {
            i++;
        }
        else if (n[i] > m[j])
        {
            j++;
        }
        else
        {
            result[resultSize++] = n[i];
            i++;
            j++;
        }
    }

    results(result, resultSize);
}

void unionNM(const int* n, int N, const int* m, int M)
{
    int result[2000], resultSize = 0;

    int i = 0, j = 0;
```

```
    int i = 0, j = 0;
    while (i < N && j < M)
    {
        if (n[i] < m[j])
        {
            result[resultSize++] = n[i++];
        }
        else if (n[i] > m[j])
        {
            result[resultSize++] = m[j++];
        }
        else
        {
            result[resultSize++] = n[i];
            i++;
            j++;
        }
    }

    while (i < N)
    {
        result[resultSize++] = n[i++];
    }

    while (j < M)
    {
        result[resultSize++] = m[j++];
    }

    results(result, resultSize);
}

void Difference(const int* n, int N, const int* m, int M)
{
    int result[2000], resultSize = 0;

    int i = 0, j = 0;
```

```
    int i = 0, j = 0;
    while (i < N && j < M)
    {
        if (n[i] < m[j])
        {
            result[resultSize++] = n[i++];
        }
        else if (n[i] > m[j])
        {
            result[resultSize++] = m[j++];
        }
        else
        {
            i++;
            j++;
        }
    }

    while (i < N)
    {
        result[resultSize++] = n[i++];
    }

    while (j < M)
    {
        result[resultSize++] = m[j++];
    }

    results(result, resultSize);
}

int main() {
    int N, M;

    cin >> N;
    int n[1000];
    for (int i = 0; i < N; i++)
```

```
    cin >> N;
    int n[1000];
    for (int i = 0; i < N; i++)
    {
        cin >> n[i];
    }

    cin >> M;
    int m[1000];
    for (int i = 0; i < M; i++)
    {
        cin >> m[i];
    }

    sort(n, n + N);
    sort(m, m + M);

    difference(n, N, m, M);
    cout << "\n";

    difference(m, M, n, N);
    cout << "\n";

    section(n, N, m, M);
    cout << "\n";

    unionNM(n, N, m, M);
    cout << "\n";

    Difference(n, N, m, M);
    cout << "\n";

    return 0;
}
```

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	Дії
декілька секунд тому	C++ 23	Зараховано	0.003	1.215	Перегляд

Lab# programming: Algotester_Lab_6 (1) (1 год)

```cpp
#include <iostream>
#include <string>
#include <algorithm>
#include <vector>
#include <map>
using namespace std;

void lowerCase(string& str) {
    int len = str.size();
    for (int letter = 0; letter < len; letter++) {
        if (str[letter] >= 65 && str[letter] <= 90)
            str[letter] += 32;
    }
}

int main() {
    int N, K;
    cin >> N >> K;
    cin.ignore(32000, '\n');
    vector <string> str(N);
    for (int i = 0; i < N; i++) {
        getline(cin, str[i]);
        lowerCase(str[i]);
    }

    map<string, int> words;
    for (const string& s : str) {
        words[s]++;
    }

    vector<bool> isPresent(26, false);
    for (const auto& pair : words) {
        if (pair.second >= K) {
            for (char ch : pair.first) {
                isPresent[ch - 'a'] = true;
            }
        }
    }

    vector<char> let;
```

```cpp
    vector<char> let;
    for (int i = 25; i >= 0; i--) {
        if (isPresent[i]) {
            let.push_back('a' + i);
        }
    }

    if (let.size() == 0)
        cout << "Empty!";
    else {
        cout << let.size() << "\n";
        for (char el : let)
            cout << el << " ";
    }
    return 0;
}
```

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	Дії
декілька секунд тому	C++ 23	Зараховано	0.058	5.859	Перегляд

Practice# programming: Class Practice_Task (2 год)

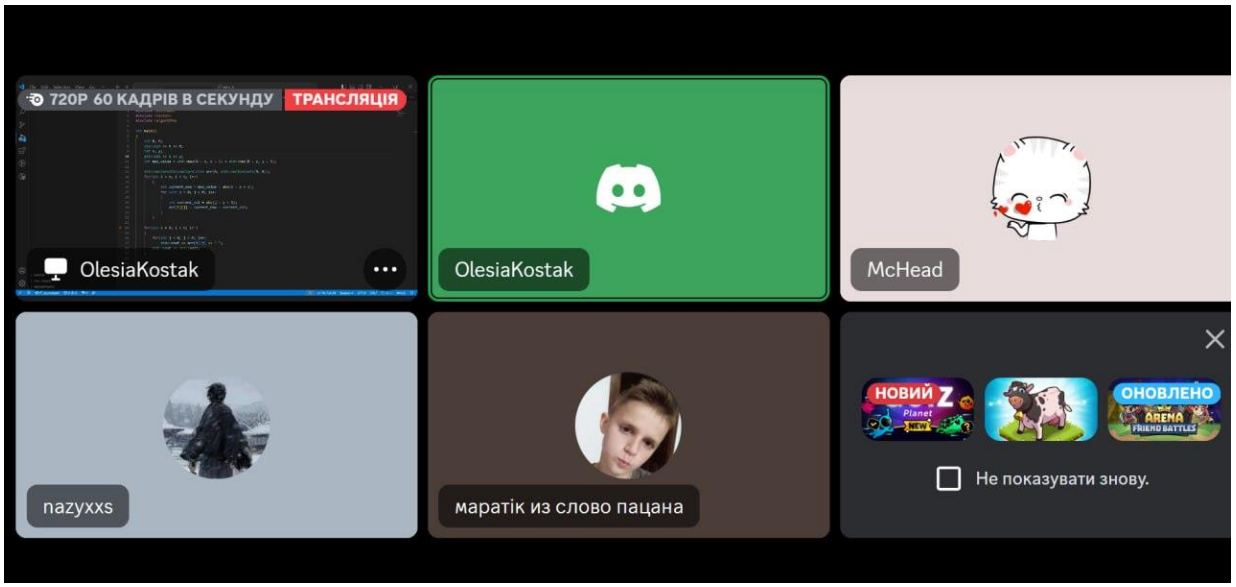
```cpp
#include <iostream>
#include <cstdio>
using namespace std;

enum FileOpResult {
    Success,
    Failure
};

FileOpResult write_to_file(char* name, char* content) {
    FILE* file = fopen(name, "w");

    if (file == nullptr) {
        return Failure;
    }

    if (fprintf(file, "%s", content) < 0) {
        fclose(file);
        return Failure;
    }

    if (fclose(file) != 0) {
        return Failure;
    }

    return Success;
}

int main() {
    char filename[256];
    char content[1024];

    cout << "Enter name of your file: ";
    cin >> filename;
    cout << "Enter text which do you wanna to see insede: ";
    cin.ignore();
    cin.getline(content, 1024);

    FileOpResult result = write_to_file(filename, content);
```

```cpp
    FileOpResult result = write_to_file(filename, content);

    if (result == Success) {
        cout << "You createdd a file" << endl;
    }
    else {
        cout << "Error " << endl;
    }

    return 0;
}
```

Practice# programming: Self Practice Task

```cpp
#include <vector>
using namespace std;

int main()
{
    int n;
    cin >> n;

    vector<int> grades(n);
    bool fail = false;
    bool excellent = true;

    for (int i = 0; i < n; ++i)
    {
        cin >> grades[i];
        if (grades[i] < 51)
        {
            fail = true;
        }
        if (grades[i] < 90)
        {
            excellent = false;
        }
    }

    if (fail)
    {
        cout << "scholarship wont be your" ;
    }
    else if (excellent)
    {
        cout << "You will have the best scholarship" ;
    }
    else
    {
        cout << "You will have scholarship" ;
    }

    return 0;
}
```

Zoom Time:



Висновок: в ході роботи над даним епіком я навчилався використовувати на практиці нові знання, такі як поняття файла, базовий файловий ввід і вивід, файли в C++, опрацювання рядків, символьний тип даних char, Літерали і магічні числа, читання і запис бінарних файлів, створення і використання бібліотек.