

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Кафедра систем штучного інтелекту



Звіт

про виконання лабораторних та практичних робіт блоку № 2

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 № 6,8,9

Алготестер Лабораторної Роботи № 4,6

Практичних Робіт до блоку № 5

Виконав:

Студент групи ШІ-11

Маркевич Владислав

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

1. Навчитись працювати з файлами у C++, зокрема створювати, відкривати, читати, записувати та закривати як текстові, так і бінарні файли.
2. Ознайомитись із використанням символів і рядкових змінних при обробці текстової інформації у файлах.
3. Засвоїти основи роботи зі стандартною бібліотекою C++ для роботи з файлами, включаючи класи `fstream`, `ifstream`, `ofstream`, та їхні методи.
4. Навчитись ефективно застосовувати деталі роботи з файлами, такі як управління позиціями вказівника читання/запису, перевірка стану файлів та обробка помилок.
5. Розробити навички створення власних бібліотек для автоматизації роботи з файлами, що забезпечують зручність і повторне використання коду в інших проектах.
6. Поглибити розуміння принципів роботи з текстовими та бінарними файлами для обробки великих обсягів даних у різних форматах.

Теоретичні відомості

- 1) Файли: <https://acode.com.ua/urok-220-bazovyj-fajlovyj-vvid-i-vyvid/>
<https://acode.com.ua/urok-221-randomnyj-fajlovyj-vvid-i-vyvid/>
- 2) Рядки: <https://acode.com.ua/urok-221-randomnyj-fajlovyj-vvid-i-vyvid/>
<https://acode.com.ua/urok-60-vvedennya-v-std-string/>
<https://acode.com.ua/urok-82-ryadky-c-style/>
- 3) Стандартна бібліотека:
<https://acode.com.ua/urok-204-standartna-biblioteka-shabloniv-stl/>
<https://acode.com.ua/urok-205-kontejnery-stl/>

<https://acode.com.ua/urok-206-iteratory-stl/>

<https://acode.com.ua/urok-207-algorytmy-stl/>

Виконання роботи

Завдання №1 VNS Lab 6 – варіант 7

```
#include <iostream>
#include <cstring>
#include <vector>
#include <algorithm>
#include <cctype>

using namespace std;

int main() {
    char s[256];

    gets(s);
    // fgets(s, sizeof(s), stdin);
    vector<char> nums;

    for (int i = 0; s[i] != '\0'; i++) {
        if (isdigit(s[i])) {
            nums.push_back(s[i]);
        }
    }
    sort(nums.begin(), nums.end(), greater<int>());

    int indexfornums = 0;
    for (int i = 0; s[i] != '\0'; i++) {
        if (isdigit(s[i])) {
            s[i] = nums[indexfornums++];
        }
    }

    cout << s << endl;

    return 0;
}
```

```
fsdfds5432gsdg66
fsdfds6654gsdg32
```

Завдання №2 VNS Lab 8 – варіант 7

```

1  #include <iostream>
2  #include <fstream>
3  #include <string>
4
5  using namespace std;
6
7  struct Student {
8      char surname[50];
9      char name[50];
10     char patronymic[50];
11     char address[100];
12     char group[20];
13     float rating;
14 };
15
16 void createFile(const char* filename) {
17     ofstream file(filename, ios::binary);
18     if (!file.is_open()) {
19         cerr << "Помилка при відкритті файлу для запису!" << endl;
20         return;
21     }
22
23     Student students[] = {
24         {"Petliura", "Olexander", "Ivanovich", "Lviv, Yevgena Lazarenko St. 42", "A1", 75.5},
25         {"Khvyliovyi", "Anton", "Petrovich", "Lviv, Levandivska St. 2", "B1", 60.2},
26         {"Hrushevskyi", "Zakhar", "Zakharovich", "Lviv, Lysenytska St. 3", "C1", 85.0},
27         {"Dovzhenko", "Serhiy", "Petrovich", "Lviv, Listova St. 4", "B1", 90.3}
28     };
29
30     file.write(reinterpret_cast<char*>(students), sizeof(students));
31     file.close();
32
33     cout << "Файл успішно створений." << endl;
34 }
35
36 void printFile(const char* filename) {
37     ifstream file(filename, ios::binary);

```

```

36 void printFile(const char* filename) {
37     if (!file.is_open()) {
38         return;
39     }
40
41     Student student;
42     while (file.read(reinterpret_cast<char*>(&student), sizeof(student))) {
43         cout << "Прізвище: " << student.surname << ", "
44             << "Ім'я: " << student.name << ", "
45             << "По батькові: " << student.patronymic << ", "
46             << "Адреса: " << student.address << ", "
47             << "Група: " << student.group << ", "
48             << "Рейтинг: " << student.rating << endl;
49     }
50     file.close();
51 }
52
53 void addStudent(const char* filename, const Student& newStudent) {
54     fstream file(filename, ios::in | ios::out | ios::binary);
55     if (!file.is_open()) {
56         cerr << "Помилка при відкритті файлу для додавання!" << endl;
57         return;
58     }
59
60     file.seekg(0, ios::end);
61     file.write(reinterpret_cast<const char*>(&newStudent), sizeof(newStudent));
62     file.close();
63
64     cout << "Студент доданий в кінець файлу." << endl;
65 }
66
67 void removeStudentsByRating(const char* filename, float minRating) {
68     ifstream file(filename, ios::binary);
69     if (!file.is_open()) {
70         cerr << "Помилка при відкритті файлу для читання!" << endl;
71         return;
72     }

```

```

77     file.seekg(0, ios::end);
78     long fileSize = file.tellg();
79     int numStudents = fileSize / sizeof(Student);
80
81     file.seekg(0, ios::beg);
82     Student* students = new Student[numStudents];
83     file.read(reinterpret_cast<char*>(students), fileSize);
84     file.close();
85
86     ofstream tempFile("temp.dat", ios::binary);
87     if (!tempFile.is_open()) {
88         cerr << "Не вдалось відкрити файл" << endl;
89         delete[] students;
90         return;
91     }
92
93     for (int i = 0; i < numStudents; i++) {
94         if (students[i].rating >= minRating) {
95             tempFile.write(reinterpret_cast<char*>(&students[i]), sizeof(Student));
96         }
97     }
98
99     tempFile.close();
100     delete[] students;
101
102     remove(filename);
103     rename("temp.dat", filename);
104
105     cout << "Студенти з рейтингом менше " << minRating << " були видалені." << endl;
106 }
107
108 int main() {
109     const char* filename = "students.dat";
110
111     createFile(filename);

```

```

int main() {
    const char* filename = "students.dat";

    createFile(filename);

    cout << "\nВміст файлу до змін:" << endl;
    printFile(filename);

    removeStudentsByRating(filename, 70.0);

    Student newStudent = {"Kovalenko", "Oleh", "Ivanovich", "Lviv, Luhansk St. 5", "B2", 80.0};
    addStudent(filename, newStudent);

    cout << "\nВміст файлу після змін:" << endl;
    printFile(filename);

    return 0;
}

```

Вміст файлу до змін:
Прізвище: Petliura, Ім'я: Olexander, По батькові: Ivanovich, Адреса: Lviv, Yevgena Lazarenko St. 42, Група: A1, Рейтинг: 75.5
Прізвище: Khyvliovyi, Ім'я: Anton, По батькові: Petrovich, Адреса: Lviv, Levandivska St. 2, Група: B1, Рейтинг: 60.2
Прізвище: Hrushevskiyi, Ім'я: Zakhar, По батькові: Zakharovich, Адреса: Lviv, Lysenytska St. 3, Група: C1, Рейтинг: 85
Прізвище: Dovzhenko, Ім'я: Serhiy, По батькові: Petrovich, Адреса: Lviv, Listova St. 4, Група: B1, Рейтинг: 90.3
Студенти з рейтингом менше 70 були видалені.
Студент доданий в кінець файлу.

Вміст файлу після змін:
Прізвище: Petliura, Ім'я: Olexander, По батькові: Ivanovich, Адреса: Lviv, Yevgena Lazarenko St. 42, Група: A1, Рейтинг: 75.5
Прізвище: Hrushevskiyi, Ім'я: Zakhar, По батькові: Zakharovich, Адреса: Lviv, Lysenytska St. 3, Група: C1, Рейтинг: 85
Прізвище: Dovzhenko, Ім'я: Serhiy, По батькові: Petrovich, Адреса: Lviv, Listova St. 4, Група: B1, Рейтинг: 90.3
Прізвище: Kovalenko, Ім'я: Oleh, По батькові: Ivanovich, Адреса: Lviv, Luhansk St. 5, Група: B2, Рейтинг: 80

Завдання №3 VNS Lab 9 – варіант 7

```
1  #include <iostream>
2  #include <fstream>
3  #include <sstream>
4  #include <string>
5
6  using namespace std;
7
8  int main() {
9      ifstream F1("F1.txt");
10
11     if (!F1.is_open()) {
12         cerr << "Не вдалося відкрити файл F1!" << endl;
13         return 1;
14     }
15
16     ofstream F2("F2.txt");
17
18     if (!F2.is_open()) {
19         cerr << "Не вдалося відкрити файл F2!" << endl;
20         return 1;
21     }
22
23     string line;
24     while (getline(F1, line)) {
25         if (!line.empty() && line[0] != 'A' && line[0] != 'a') {
26             F2 << line << endl;
27         }
28     }
29     F1.close();
30     F2.close();
31
32     ifstream F2_read("F2.txt");
33
34     if (!F2_read.is_open()) {
35         cerr << "Не вдалося відкрити файл F2!" << endl;
36         return 1;
37     }
```

```

if (!F2_read.is_open()) {
    cerr << "Не вдалося відкрити файл F2!" << endl;
    return 1;
}

string first_word;

if (getline(F2_read, line)) {
    stringstream ss(line);
    ss >> first_word;
}

cout << "Кількість символів у першому слові F2: " << first_word.length() << endl;

F2_read.close();

return 0;
}

```

Кількість символів у першому слові F2: 5

Завдання №4 Algotester Lab 4v3


```
#include <stdio.h>
#include <iostream>

using namespace std;

void sort1(int arr[], int n) {
    for (int i = 0; i < n - 1; i++) {
        for (int j = i + 1; j < n; j++) {
            if (arr[i] < arr[j]) {
                int temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }
}

void sort2(int arr[], int n) {
    for (int i = 0; i < n - 1; i++) {
        for (int j = i + 1; j < n; j++) {
            if (arr[i] > arr[j]) {
                int temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }
}

int removeDuplicates(int arr[], int n) {
    if (n == 0 || n == 1) {
        return n;
    }

    int j = 0;
    for (int i = 0; i < n - 1; i++) {
        if (arr[i] != arr[i + 1]) {
```

```
45  ∨ int main() {
46      int N;
47      scanf("%d", &N);
48
49      int *array = new int[N];
50
51  ∨  for (int i = 0; i < N; i++) {
52      |   scanf("%d", &array[i]);
53      |   }
54
55      int *remainder0 = new int[N];
56      int *remainder1 = new int[N];
57      int *remainder2 = new int[N];
58      int count0 = 0, count1 = 0, count2 = 0;
59
60  ∨  for (int i = 0; i < N; i++) {
61  ∨      |   if (array[i] % 3 == 0) {
62      |       remainder0[count0++] = array[i];
63  ∨      |   } else if (array[i] % 3 == 1) {
64      |       remainder1[count1++] = array[i];
65  ∨      |   } else {
66      |       remainder2[count2++] = array[i];
67      |   }
68      |   }
69
70      sort2(remainder0, count0);
71      sort1(remainder1, count1);
72      sort2(remainder2, count2);
73      int *result = new int[N];
74      int index = 0;
75  ∨  for (int i = 0; i < count0; i++) {
76      |   result[index++] = remainder0[i];
77      |   }
78
79  ∨  for (int i = 0; i < count1; i++) {
```

```

76         result[index++] = remainder0[i];
77     }
78
79     for (int i = 0; i < count1; i++) {
80         result[index++] = remainder1[i];
81     }
82
83     for (int i = 0; i < count2; i++) {
84         result[index++] = remainder2[i];
85     }
86
87     int new_size = removeDuplicates(result, index);
88
89     printf("%d\n", new_size);
90     for (int i = 0; i < new_size; i++) {
91         printf("%d ", result[i]);
92     }
93     printf("\n");
94
95     delete[] array;
96     delete[] remainder0;
97     delete[] remainder1;
98     delete[] remainder2;
99     delete[] result;
100
101     return 0;
102 }

```

10

1 33 4 8 6 5 2 7 5 0

9

0 6 33 7 4 1 2 5 8

```

#include <stdio.h>
#include <algorithm>
#include <iostream>

using namespace std;

int main() {
    int N;
    scanf("%d", &N);

    int *array = new int[N];

    for (int i = 0; i < N; i++) {
        scanf("%d", &array[i]);
    }

    int* zero = partition(array, array + N, [](int i) { return i % 3 == 0; });
    int* one = partition(zero, array + N, [](int i) { return i % 3 == 1; });

    sort(array, zero);
    sort(zero, one, [](int a, int b) {
        return a > b;});
    sort(one, array + N);

    int *end = unique(array, array + N);
    int new_size = distance(array, end);

    printf("%d\n", new_size);
    for (int i = 0; i < new_size; i++) {
        printf("%d ", array[i]);
    }

    delete[] array;

    return 0;
}

```

```

10
1 33 4 8 6 5 2 7 5 0
9
0 6 33 7 4 1 2 5 8

```

Завдання №5 Algotester Lab 6v7

```

1  #include <iostream>
2  #include <string>
3  #include <algorithm>
4  #include <vector>
5  #include <map>
6  using namespace std;
7
8  void lowerCase(string &str) {
9      int len = str.size();
10     for(int letter = 0; letter < len; letter++) {
11         if(str[letter] >= 65 && str[letter] <= 90)
12             str[letter] += 32;
13     }
14 }
15
16 int main() {
17     int N, K;
18     cin >> N >> K;
19     cin.ignore(32000, '\n');
20     vector<string> str(N);
21     for(int i = 0; i < N; i++) {
22         getline(cin, str[i]);
23         lowerCase(str[i]);
24     }
25
26     map<string, int> words;
27     for (const string &s : str) {
28         words[s]++;

```

```

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

    if(let.size()==0)
        cout << "Empty!";
    else {

```

```

48         cout << "Empty! ";
49     else {
50         cout << let.size() << "\n";
51         for(char el : let)
52             cout << el << " ";
53     }
54     return 0;
55 }

```

```

5 2
stugna
neptune
grim
oplot
Grim
4
r m i g

```

Завдання №6 Class Practice Work

```

1  ∨ #include <iostream>
2  ∨ #include <fstream>
3  ∨ #include <string>
4  using namespace std;
5
6  enum FileOpResult { Success, Failure };
7
8  ∨ FileOpResult write_to_file(const char *name,const char *content){
9  ∨     if(name == NULL || content == NULL){
10         cout<<"Неправильна назва файлу або вмісту!"<<endl;
11         return Failure;
12     }
13
14     ofstream file(name);
15
16     ∨ if(!file.is_open()){
17         cout<<"Не вдалось відкрити файл!"<<endl;
18         return Failure;
19     }
20
21     file<<content;
22     ∨ if(!file){
23         cout<<"Помилка запису"<<endl;
24         file.close();
25         return Failure;
26     }
27
28     file.close();
29
30     ∨ if(file.fail()){
31         cout<<"Помилка закриття файлу"<<endl;
32         return Failure;
33     }else{

```

```

file.close();
if(file.fail()){
    cout<<"Помилка закриття файлу"<<endl;
    return Failure;
}else{
    return Success;
}
}
int main(){
    char filename[1024];
    char content[1024];
    cin.getline(filename,1024);
    cin.getline(content,1024);
    FileOpResult result = write_to_file(filename, content);

    if(result == Success){
        cout<<"Все записано успішно!"<<endl;
    }
    else{
        cout<<"На жаль, не вдалось записати зміст!"<<endl;
    }
    return 0;
}

```

```

Try1.txt
Hello
Hello
Все записано успішно!

```

```

1  #include <iostream>
2  #include <fstream>
3  #include <string>
4
5  using namespace std;
6
7  enum FileOpResult { Success, Failure};
8  FileOpResult copy_file(const char *file_from, const char *file_to){
9      if(file_from == NULL || file_to == NULL){
10         cout<<"Щось пішло не так!"<<endl;
11         return Failure;
12     }
13
14     ifstream fileto(file_from, ios::binary);
15     if(!fileto.is_open()){
16         cout<<"Не вдалось відкрити файл!"<<endl;
17         return Failure;
18     }
19
20     ofstream filefrom(file_to, ios::binary);
21     if(!filefrom.is_open()){
22         cout<<"Не вдалось відкрити файл!"<<endl;
23         return Failure;
24     }
25
26
27     filefrom << fileto.rdbuf();
28
29     if(!filefrom){
30         cout<<"Не вдалось прочитати файл!"<<endl;
31         return Failure;
32     }

```

```

    if(!filefrom){
        cout<<"Не вдалось прочитати файл!"<<endl;
        return Failure;
    }

    fileto.close();
    filefrom.close();
    return Success;
}

int main(){
    char filein[1024];
    char fileout[1024];

    cin.getline(filein, 1024);
    cin.getline(fileout, 1024);

    FileOpResult finalresult = copy_file (filein, fileout);
    if(finalresult == Success){
        cout<<"Все записано успішно!"<<endl;
    }
    else{
        cout<<"На жаль, не вдалось записати зміст!"<<endl;
    }

    return 0;
}

```



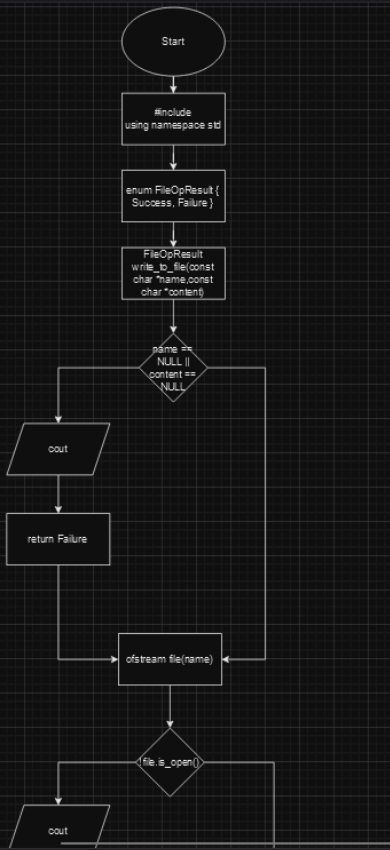
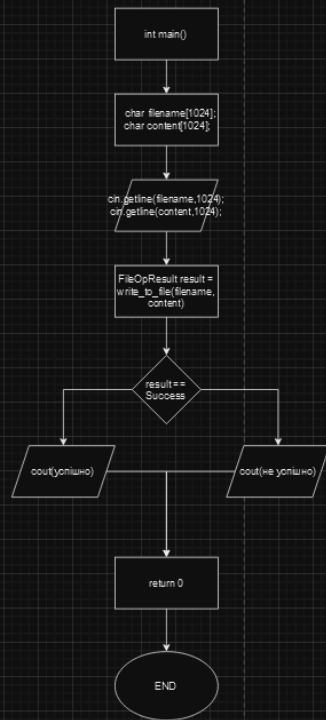
```
Try1.txt
Try2.txt
Все записано успішно!
```

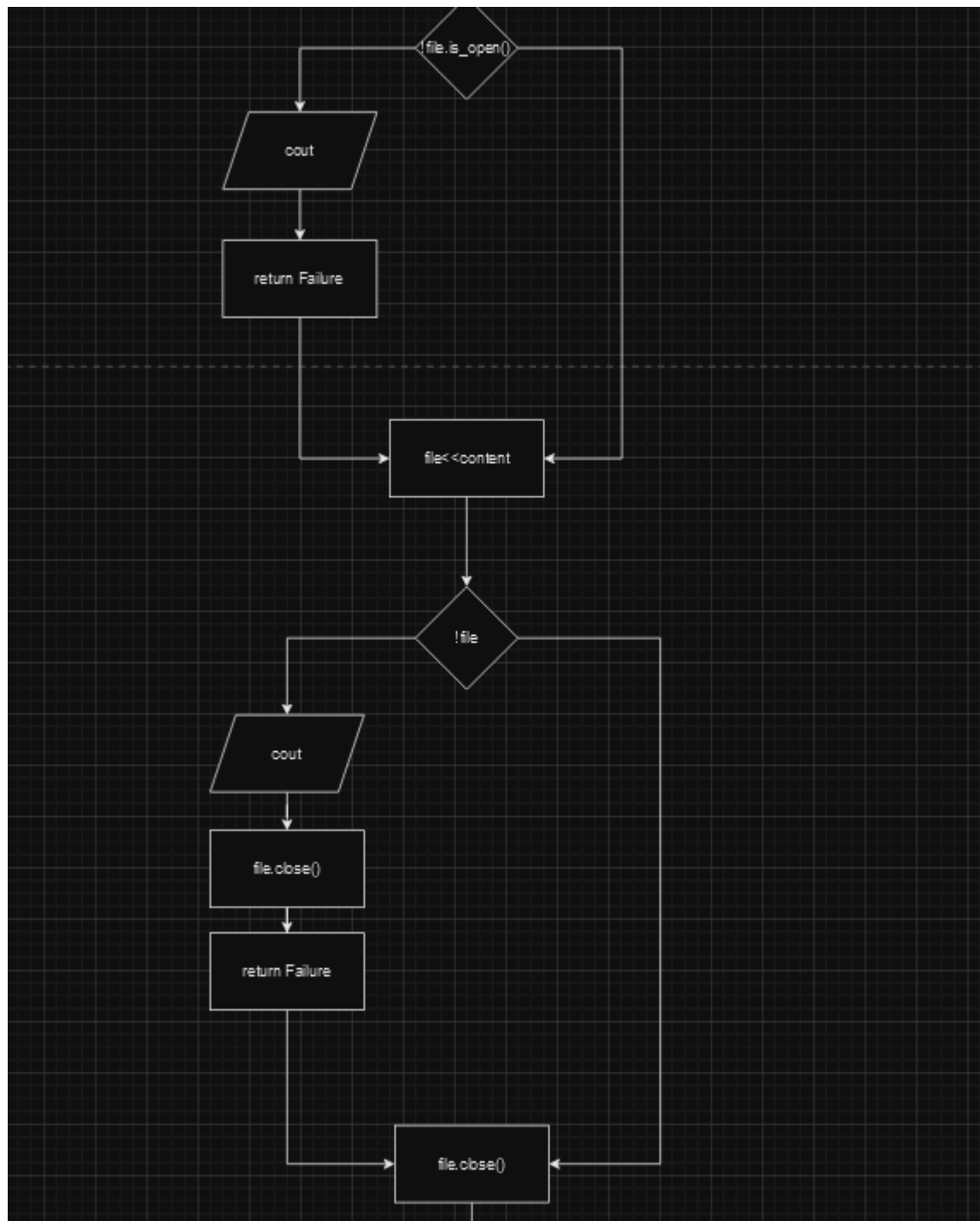
Завдання №7 Self Practice Work

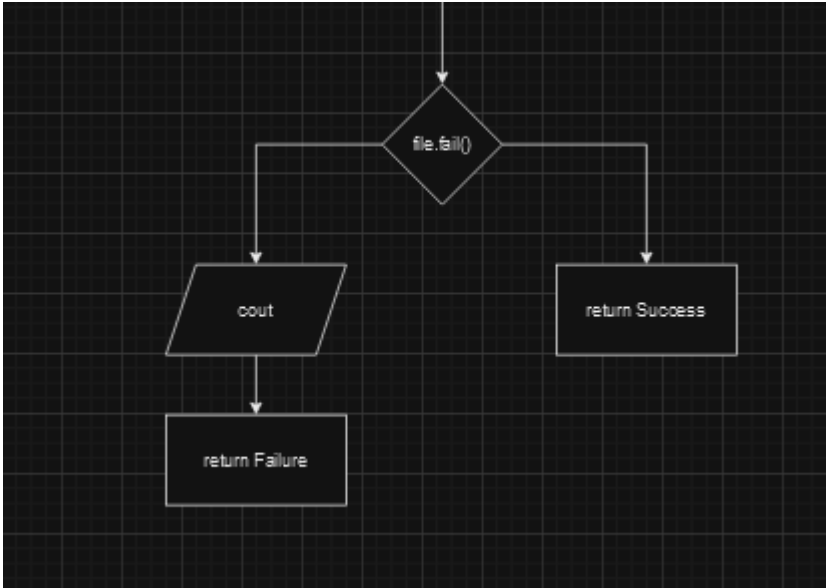
```
1  #include <iostream>
2  #include <cmath>
3
4  using namespace std;
5
6  int main(){
7      double l,w,u,d;
8      cin>>l>>w>>u>>d;
9
10     if(u+d >=l && w>=l){
11         cout<<"Three times Sex on the Beach, please!"<<endl;
12     }else{
13         cout<<"Forget about the cocktails, man!"<<endl;
14     }
15
16     return 0;
17 }
```

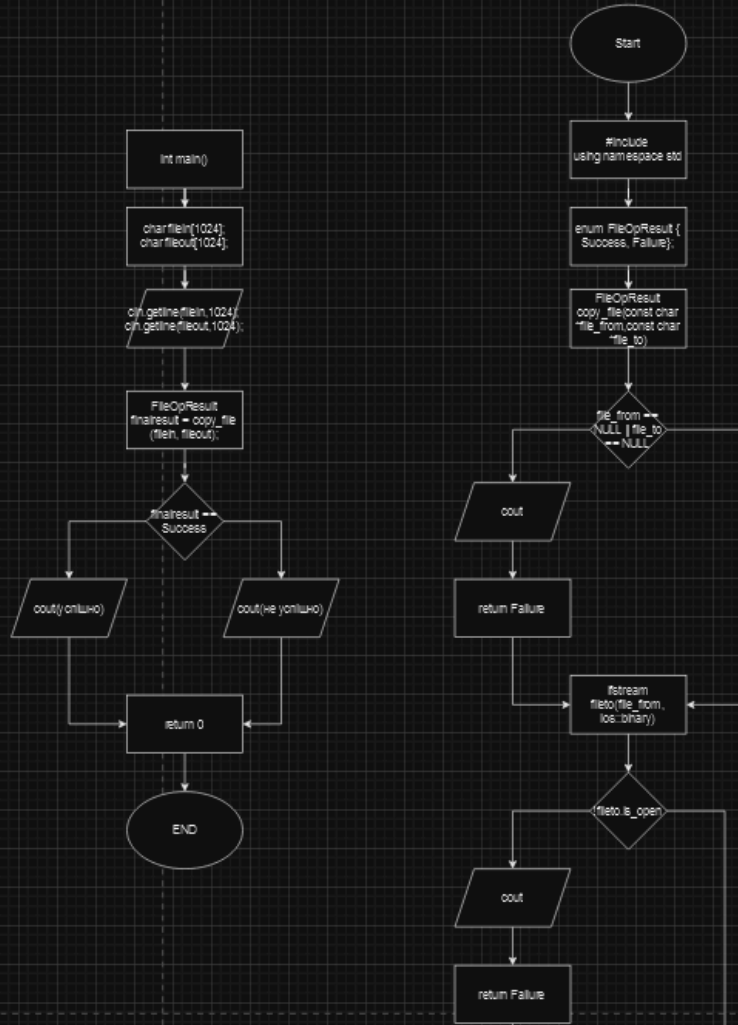
```
7 11 4 4
Three times Sex on the Beach, please!
```

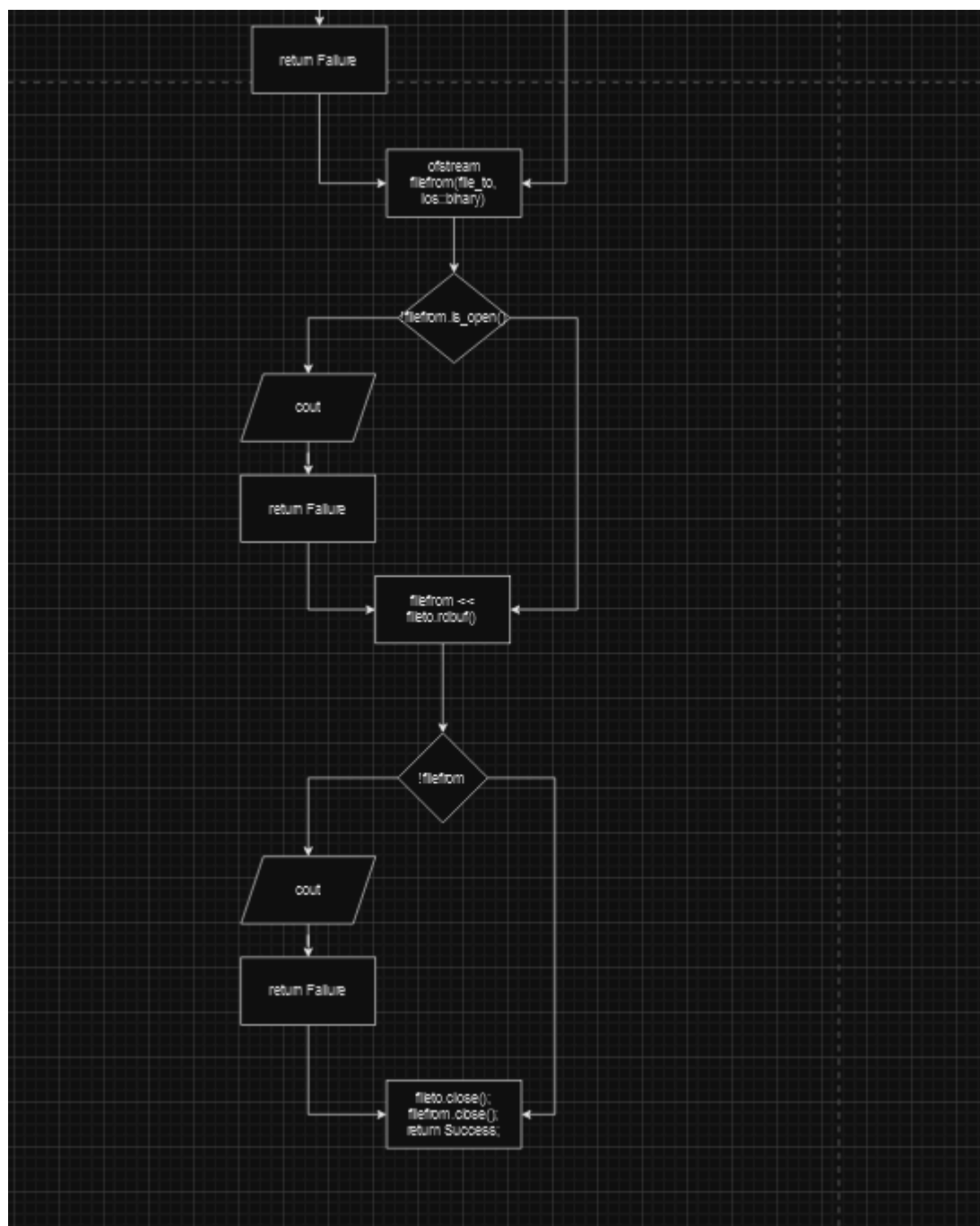
Блок-схема:



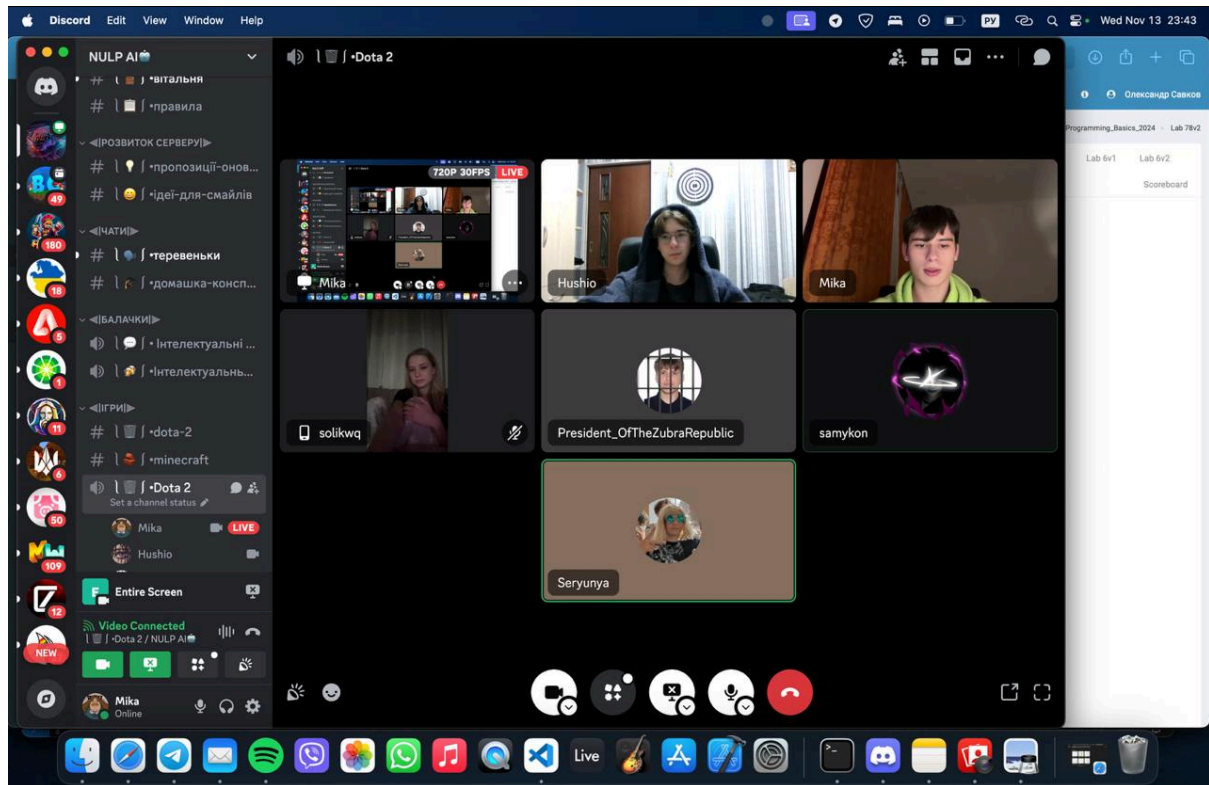








Зустріч з командою



Висновок

Під час виконання цього епіку я навчився працювати з файлами у C++, включаючи створення, відкриття, читання, запис і закриття текстових і бінарних файлів. Опанував використання класів стандартної бібліотеки (fstream, ifstream, ofstream), управління позиціями вказівника читання/запису, перевірку стану файлів та обробку помилок. Також вивчив кілька бібліотечних функцій для ефективнішої роботи з масивами.