

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-12

Токарік Сергій

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 проєктах.
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 24

```
1  #include <stdio.h>
2  #include <stdbool.h>
3  #include <string.h>
4
5  int countWords(const char *str) {
6      int count = 0;
7      bool inWord = false;
8
9      for (int i = 0; str[i] != '\0'; i++) {
10         if (str[i] != ' ' && str[i] != '.') {
11             if (!inWord) {
12                 count++;
13                 inWord = true;
14             }
15         } else {
16             inWord = false;
17         }
18     }
19
20     return count;
21 }
22
23 int main() {
24     char s[256];
25     //gets(s);
26     printf("Введіть текст (не більше 255 символів, закінчується крапкою): ");
27     if (fgets(s, sizeof(s), stdin) == NULL) {
28         printf("Помилка введення!\n");
29         return 1;
30     }
31
32     size_t len = strlen(s);
33     if (len > 0 && s[len - 1] == '\n') {
34         s[len - 1] = '\0';
35     }
36
37     int wordCount = countWords(s);
38     printf("Кількість слів у рядку: %d\n", wordCount);
39
40     return 0;
41 }
```

Введіть текст (не більше 255 символів, закінчується крапкою): fkff kfkfj kffjf jfjfjf.
Кількість слів у рядку: 4

Завдання №2 VNS Lab 8 – варіант 24

```

1 #include <iostream>
2 #include <fstream>
3 #include <vector>
4 #include <string>
5 #include <algorithm>
6
7 using namespace std;
8
9 struct StudentInfo {
10     string full_name;
11     string group_name;
12     string phone_number;
13     int subject_marks[3];
14 };
15
16 void addNewStudent(vector<StudentInfo>& student_list) {
17     StudentInfo new_student;
18     cout << "Введіть ім'я студента: ";
19     getline(cin, new_student.full_name);
20     cout << "Введіть номер телефону: ";
21     getline(cin, new_student.phone_number);
22     cout << "Введіть групу: ";
23     getline(cin, new_student.group_name);
24     cout << "Введіть оцінки по 3 предметах: ";
25     cin >> new_student.subject_marks[0] >> new_student.subject_marks[1] >> new_student.subject_marks[2];
26     cin.ignore(); // очищаємо буфер вводу
27     student_list.push_back(new_student);
28 }
29
30 void deleteStudentIfLowGrades(vector<StudentInfo>& student_list, const string& group, float min_average) {
31     auto it = student_list.begin();
32     while (it != student_list.end()) {
33         float avg_marks = (it->subject_marks[0] + it->subject_marks[1] + it->subject_marks[2]) / 3.0f;
34         if (it->group_name == group && avg_marks < min_average) {
35             it = student_list.erase(it);
36         } else {
37             ++it;
38         }
39     }
40 }
41
42 void printStudents(const vector<StudentInfo>& student_list) {
43     for (const auto& student : student_list) {
44         cout << "Ім'я: " << student.full_name << endl;
45         cout << "Телефон: " << student.phone_number << endl;
46         cout << "Група: " << student.group_name << endl;
47         cout << "Оцінки: " << student.subject_marks[0] << ", "
48             << student.subject_marks[1] << ", "
49             << student.subject_marks[2] << endl;
50         cout << "-----" << endl;
51     }
52 }
53
54 void saveToBinaryFile(const vector<StudentInfo>& student_list, const string& filename) {
55     ofstream file(filename, ios::binary);
56     if (!file) {
57         cerr << "Не вдалося відкрити файл для запису!" << endl;
58         return;
59     }
60     for (const auto& student : student_list) {
61         file.write(student.full_name.c_str(), student.full_name.size() + 1);
62         file.write(student.phone_number.c_str(), student.phone_number.size() + 1);
63         file.write(student.group_name.c_str(), student.group_name.size() + 1);
64         file.write(reinterpret_cast<const char*>(student.subject_marks), sizeof(student.subject_marks));
65     }
66     file.close();
67 }

```

```

54 void saveToBinaryFile(const vector<StudentInfo>& student_list, const string& filename) {
55     file.close();
56 }
57
58
59 void loadFromFile(vector<StudentInfo>& student_list, const string& filename) {
60     ifstream file(filename, ios::binary);
61     if (!file) {
62         cerr << "Не вдалося відкрити файл для читання!" << endl;
63         return;
64     }
65
66     StudentInfo student;
67     while (getline(file, student.full_name, '\0')) {
68         getline(file, student.phone_number, '\0');
69         getline(file, student.group_name, '\0');
70         file.read(reinterpret_cast<char*>(student.subject_marks), sizeof(student.subject_marks));
71         student_list.push_back(student);
72     }
73
74     file.close();
75 }
76
77 int main() {
78     vector<StudentInfo> students;
79     string filename = "students.dat";
80     int action;
81
82     while (true) {
83         cout << "1. Додати студента\n"
84              << "2. Видалити студентів за середнім балом\n"
85              << "3. Показати список студентів\n"
86              << "4. Зберегти дані в файл\n"
87              << "5. Завантажити дані з файлу\n"
88              << "6. Вийти\n"
89              << "Ваш вибір: ";
90         cin >> action;
91         cin.ignore(); // Очищаємо буфер
92
93         switch (action) {
94             case 1: {
95                 addNewStudent(students);
96                 break;
97             }
98             case 2: {
99                 string group;
100                 float minAvg;
101                 cout << "Введіть групу: ";
102                 getline(cin, group);
103                 cout << "Введіть мінімальний середній бал: ";
104                 cin >> minAvg;
105                 cin.ignore();
106                 deleteStudentIfLowGrades(students, group, minAvg);
107                 break;
108             }
109             case 3: {
110                 printStudents(students);
111                 break;
112             }
113             case 4: {
114                 saveToBinaryFile(students, filename);
115                 break;
116             }
117             case 5: {
118                 loadFromFile(students, filename);
119                 break;
120             }
121             case 6: {
122                 cout << "Вихід з програми." << endl;
123                 return 0;
124             }
125             default: {
126                 cout << "Невірний вибір. Спробуйте ще раз.\n";
127                 break;
128             }
129         }
130     }
131
132     return 0;
133 }

```

```

124         break;
125     case 5:
126         loadFromFile(students, filename);
127         break;
128     case 6:
129         cout << "Вихід з програми." << endl;
130         return 0;
131     default:
132         cout << "Невірний вибір. Спробуйте ще раз.\n";
133         break;
134     }
135 }
136
137 return 0;
138 }
139

```

```
6. Вийти
Ваш вибір: 1
Введіть ім'я студента: lsdkdk ddkdj
Введіть номер телефону: 05505050
Введіть групу: CC-22
Введіть оцінки по 3 предметах: 10 10 10
1. Додати студента
2. Видалити студентів за середнім балом
3. Показати список студентів
4. Зберегти дані у файл
5. Завантажити дані з файлу
6. Вийти
Ваш вибір: 3
Ім'я: lsdkdk ddkdj
Телефон: 05505050
Група: CC-22
Оцінки: 10, 10, 10
-----
1. Додати студента
2. Видалити студентів за середнім балом
3. Показати список студентів
4. Зберегти дані у файл
5. Завантажити дані з файлу
6. Вийти
Ваш вибір: █
```

```

1  #include <iostream>
2  #include <fstream>
3  #include <string>
4  #include <sstream>
5  #include <vector>
6
7  using namespace std;
8
9  // Функція для перевірки наявності однакових слів у рядку
10 bool containsDuplicateWords(const string& line) {
11     vector<string> words;
12     stringstream ss(line);
13     string word;
14
15     while (ss >> word) {
16         for (const auto& w : words) {
17             if (w == word) {
18                 return true;
19             }
20         }
21         words.push_back(word);
22     }
23     return false;
24 }
25
26 // Функція для пошуку індексу слова з найбільшою кількістю цифр
27 int findWordWithMostDigits(const string& line) {
28     stringstream ss(line);
29     string word;
30     int maxDigits = 0;
31     int wordIndex = -1;
32     int currentIndex = 0;
33
34     while (ss >> word) {
35         int digitCount = 0;
36         for (char c : word) {
37             if (isdigit(c)) {
38                 digitCount++;
39             }
40         }
41         if (digitCount > maxDigits) {
42             maxDigits = digitCount;
43             wordIndex = currentIndex;
44         }
45         ++currentIndex;
46     }
47     return wordIndex;
48 }
49
50 int main() {
51     ifstream input("F1.txt");
52     ofstream output("F2.txt");
53
54     if (!input) {
55         cerr << "Не вдалося відкрити вхідний файл." << endl;

```

```

55     cerr << "Не вдалося відкрити вхідний файл." << endl;
56     return 1;
57 }
58
59 if (!output) {
60     cerr << "Не вдалося відкрити вихідний файл." << endl;
61     return 1;
62 }
63
64 string line;
65 while (getline(input, line)) {
66     // Копіюємо рядки з дублікатами слів
67     if (containsDuplicateWords(line)) {
68         output << line << endl;
69     }
70 }
71
72 input.clear(); // Очистити флаг помилки після завершення роботи з файлом
73 input.seekg(0, ios::beg); // Повернутися на початок файлу для наступного проходу
74
75 cout << "Номер слова з найбільшою кількістю цифр у кожному рядку:" << endl;
76 while (getline(input, line)) {
77     int index = findWordWithMostDigits(line);
78     if (index == -1) {
79         cout << "У рядку немає слів з цифрами" << endl;
80     } else {
81         cout << "Номер слова: " << index << endl;
82     }
83 }
84
85 return 0;
86 }
87

```

```

Номер слова з найбільшою кількістю цифр у кожному рядку:
Номер слова: 11
У рядку немає слів з цифрами
У рядку немає слів з цифрами
У рядку немає слів з цифрами

```

```

≡ F1.txt
1  bro bro bro bro  hello how are you 1 23 56 565676.
2  Hey papa.
3  bambam yigfkf.
4  fjfdhf.

```

```

≡ F2.txt
1  bro bro bro bro  hello how are you 1 23 56 565676.
2

```



```

1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4
5  using namespace std;
6
7  int main() {
8      int N, K;
9      cin >> N >> K;
10     vector<int> arr(N);
11
12     for (int i = 0; i < N; i++) {
13         cin >> arr[i];
14     }
15
16     sort(arr.begin(), arr.end());
17     auto last = unique(arr.begin(), arr.end());
18     arr.erase(last, arr.end());
19
20     int n = arr.size();
21
22     K %= n;
23     rotate(arr.begin(), arr.begin() + K, arr.end());
24
25     cout << n << endl;
26     for (int num : arr) {
27         cout << num << " ";
28     }
29     cout << endl;
30
31     return 0;
32 }

```

```

10 3
1 2 2 3 3 3 4 5 6 7
7
4 5 6 7 1 2 3

```

```

1  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5
6  vector<int> remove_duplicates(const vector<int>& arr) {
7      vector<int> unique_arr;
8      vector<bool> seen(101, false);
9
10     for (int num : arr) {
11         if (!seen[num]) {
12             seen[num] = true;
13             unique_arr.push_back(num);
14         }
15     }
16
17     for (int i = 0; i < unique_arr.size(); i++) {
18         for (int j = i + 1; j < unique_arr.size(); j++) {
19             if (unique_arr[i] > unique_arr[j]) {
20                 swap(unique_arr[i], unique_arr[j]);
21             }
22         }
23     }
24     return unique_arr;
25 }
26
27 void rotate_array(vector<int>& arr, int k) {
28     int n = arr.size();
29     k %= n;
30     vector<int> temp(n);
31
32     for (int i = 0; i < n; i++) {
33         temp[i] = arr[(i + k) % n];
34     }
35     arr = temp;
36 }
37
38 int main() {
39     int N, K;
40     cin >> N >> K;
41
42     vector<int> a(N);
43     for (int i = 0; i < N; i++) {
44         cin >> a[i];
45     }
46
47     vector<int> unique_arr = remove_duplicates(a);
48
49     int n = unique_arr.size();
50
51     rotate_array(unique_arr, K);
52
53     cout << n << endl;
54     for (int x : unique_arr) {
55         cout << x << " ";
56     }
57     cout << endl;
58
59     return 0;
60 }

```

```

10 3
1 2 2 3 3 3 4 5 6 7
7
4 5 6 7 1 2 3

```

Завдання №5 Algotester Lab 6v3

```

1 #include <iostream>
2 #include <vector>
3 #include <set>
4 #include <string>
5
6 using namespace std;
7
8 vector<int> findNumbers(const vector<string> &sudoku, int x, int y) {
9     vector<int> possibleNumbers;
10    int N = sudoku.size();
11    if (sudoku[x][y] != '0') {
12        possibleNumbers.push_back(sudoku[x][y] - '0');
13        return possibleNumbers;
14    }
15    set<int> usedNumbers;
16    for (int i = 0; i < N; i++) {
17        if (sudoku[x][i] != '0') usedNumbers.insert(sudoku[x][i] - '0');
18        if (sudoku[i][y] != '0') usedNumbers.insert(sudoku[i][y] - '0');
19    }
20    for (int num = 1; num <= N; num++) {
21        if (usedNumbers.find(num) == usedNumbers.end()) {
22            possibleNumbers.push_back(num);
23        }
24    }
25    return possibleNumbers;
26 }
27
28 int main() {
29     int N;
30     cin >> N;
31     vector<string> sudoku(N);
32     for (int i = 0; i < N; i++) {
33         cin >> sudoku[i];
34     }
35     int Q;
36     cin >> Q;
37     vector<pair<int, int>> queries(Q);
38
39     for (int i = 0; i < Q; i++) {
40         int x, y;
41         cin >> x >> y;
42         queries[i] = {x - 1, y - 1};
43     }
44
45     for (const auto &query : queries) {
46         int x = query.first;
47         int y = query.second;
48         vector<int> result = findNumbers(sudoku, x, y);
49         cout << result.size() << endl;
50         for (int num : result) {
51             cout << num << " ";
52         }
53         cout << endl;
54     }
55     return 0;
56 }

```

```

3
000
100
003
3
1 1
2 3
2 1
2
2 3
1
2
1
1

```

Завдання №6 Class Practice Work

```

1  #include <iostream>
2  #include <fstream>
3  #include <string>
4
5  using namespace std;
6
7  enum FileOpResult { Success, Failure };
8
9
10 FileOpResult write_to_file(const char *name, const char *content) {
11     if (name == NULL || content == NULL) {
12         cout << "Неправильна назва файлу або вмісту!" << endl;
13         return Failure;
14     }
15
16     ofstream file(name);
17
18     if (!file.is_open()) {
19         cout << "Не вдалось відкрити файл!" << endl;
20         return Failure;
21     }
22
23     file << content;
24     if (!file) {
25         cout << "Помилка запису" << endl;
26         file.close();
27         return Failure;
28     }
29
30     file.close();
31     if (file.fail()) {
32         cout << "Помилка закриття файлу" << endl;
33         return Failure;
34     } else {
35         return Success;
36     }
37 }
38
39
40 FileOpResult copy_file(const char *file_from, const char *file_to) {
41     if (file_from == NULL || file_to == NULL) {
42         cout << "Щось пішло не так!" << endl;
43         return Failure;
44     }
45
46     ifstream file_in(file_from, ios::binary);
47     if (!file_in.is_open()) {
48         cout << "Не вдалось відкрити файл!" << endl;
49         return Failure;
50     }
51
52     ofstream file_out(file_to, ios::binary);
53     if (!file_out.is_open()) {
54         cout << "Не вдалось створити файл!" << endl;
55         return Failure;
56     }
57
58     file_out << file_in.rdbuf();
59
60     if (!file_out) {
61         cout << "Не вдалось записати файл!" << endl;
62         return Failure;
63     }
64
65     file_in.close();
66     file_out.close();
67     return Success;
68 }

```

```

67     return Success;
68 }
69
70 int main() {
71     int choice;
72     cout << "Оберіть операцію:\n1 - Записати файл\n2 - Скопіювати файл\nВаш вибір: ";
73     cin >> choice;
74     cin.ignore(); // Очищення буфера вводу
75
76     if (choice == 1) {
77         char filename[1024];
78         char content[1024];
79
80         cout << "Введіть назву файлу: ";
81         cin.getline(filename, 1024);
82         cout << "Введіть вміст для запису: ";
83         cin.getline(content, 1024);
84
85         FileOpResult result = write_to_file(filename, content);
86
87         if (result == Success) {
88             cout << "Все записано успішно!" << endl;
89         } else {
90             cout << "На жаль, не вдалось записати вміст! :(" << endl;
91         }
92     } else if (choice == 2) {
93         char filein[1024];
94         char fileout[1024];
95
96         cout << "Введіть назву файлу-джерела: ";
97         cin.getline(filein, 1024);
98         cout << "Введіть назву файлу-призначення: ";
99         cin.getline(fileout, 1024);
100
101         FileOpResult finalresult = copy_file(filein, fileout);
102
103         if (finalresult == Success) {
104             cout << "Файл успішно скопійовано!" << endl;
105         } else {
106             cout << "На жаль, не вдалось скопіювати файл! :(" << endl;
107         }
108     } else {
109         cout << "Неправильний вибір!" << endl;
110     }
111
112     return 0;
113 }

```

```

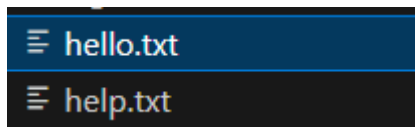
Оберіть операцію:
1 - Записати у файл
2 - Скопіювати файл
Ваш вибір: 1
Введіть назву файлу: hello.txt
Введіть вміст для запису: hello bro
Все записано успішно!

```

```

Оберіть операцію:
1 - Записати у файл
2 - Скопіювати файл
Ваш вибір: 2
Введіть назву файлу-джерела: hello.txt
Введіть назву файлу-призначення: help.txt
Файл успішно скопійовано!

```

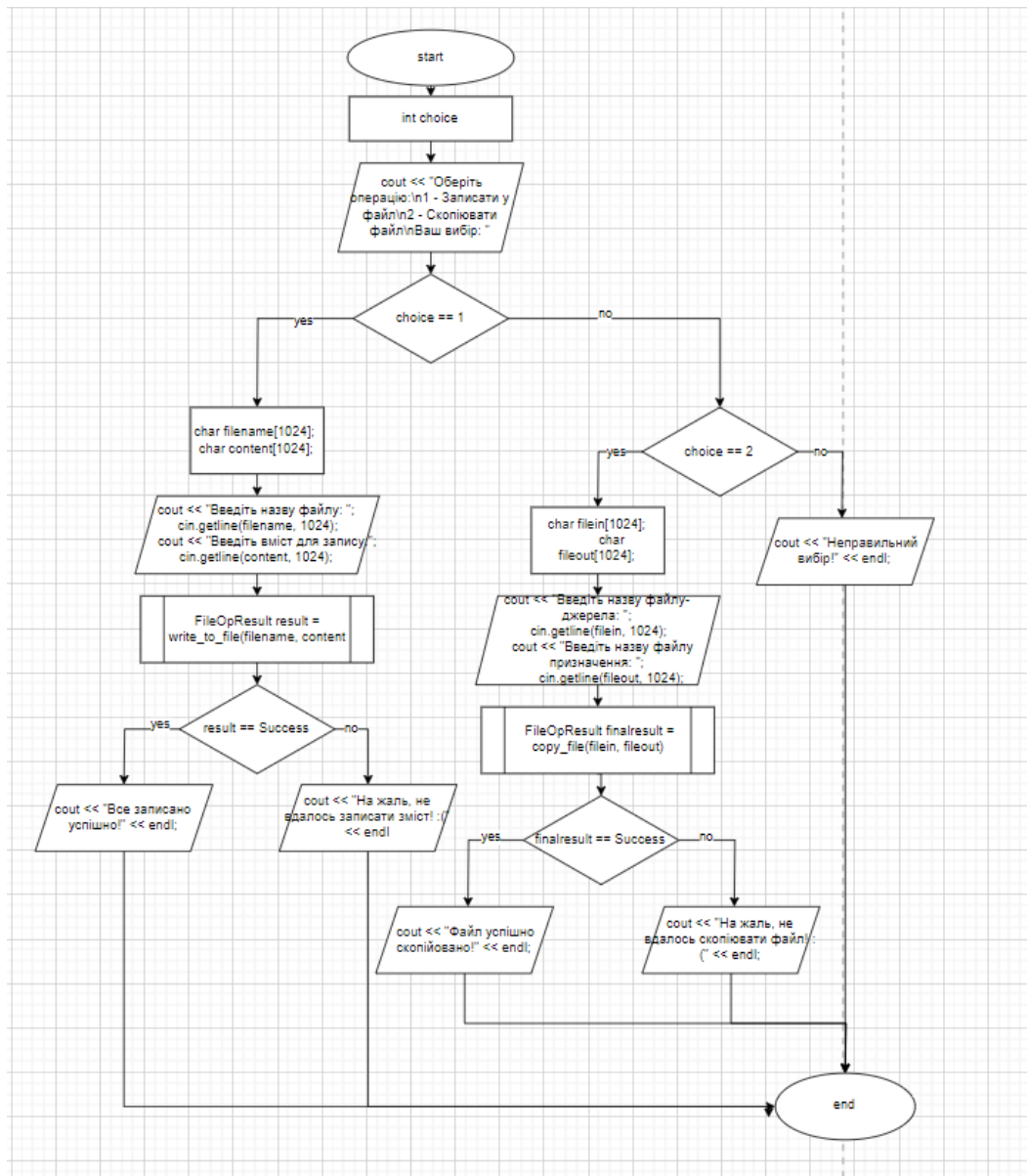


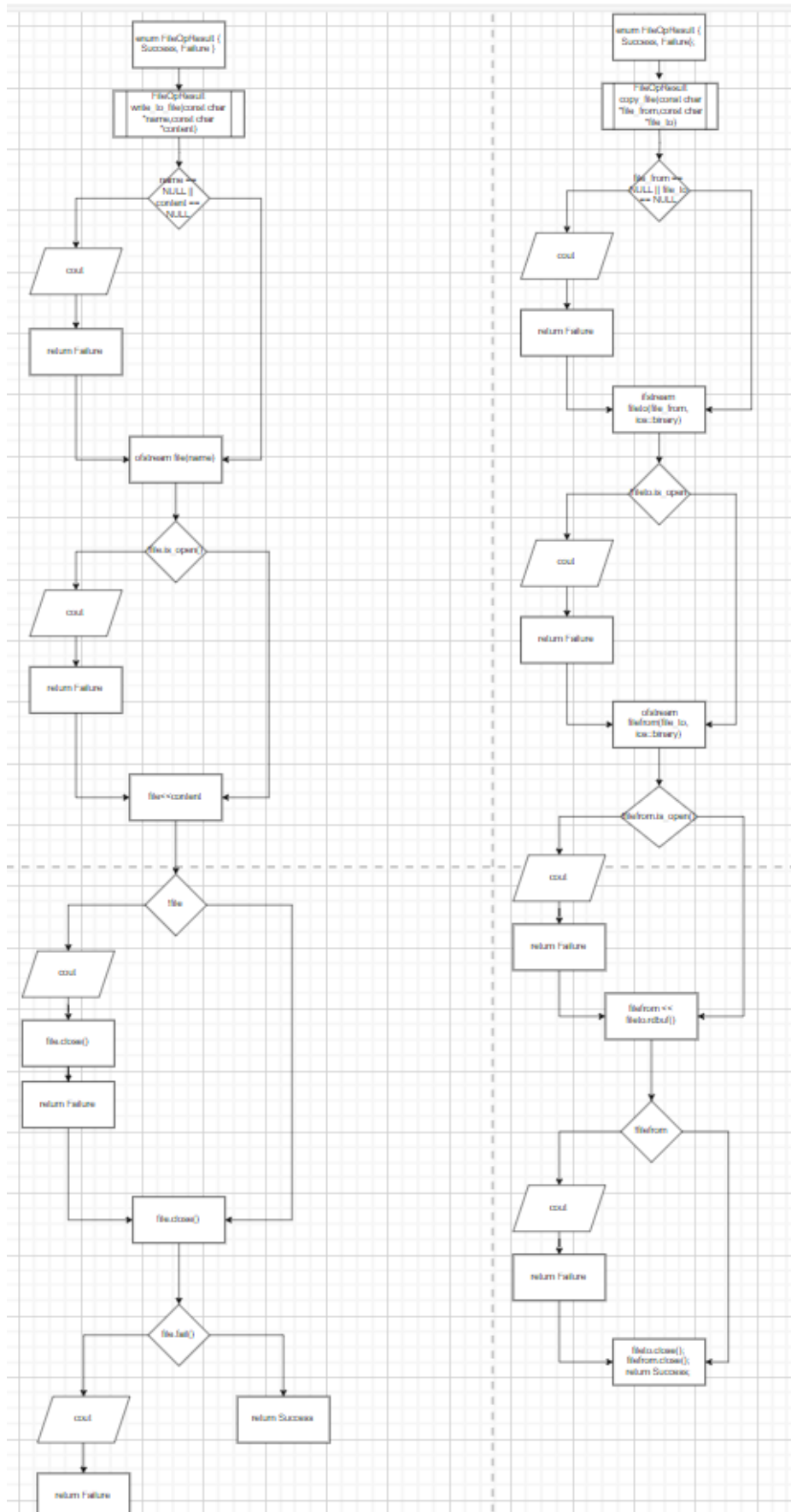
Завдання №7 Self Practice Work

```
1  #include <iostream>
2  #include <cmath>
3  #include <string>
4
5  using namespace std;
6
7  int main()
8  {
9      int n;
10     cin >> n;
11
12     string score;
13     cin >> score;
14
15     int gamescoreK = 0;
16     int gamescoreV = 0;
17     int scoreK = 0;
18     int scoreV = 0;
19
20     for(char result : score)
21     {
22         if(result == 'K')
23         {
24             scoreK++;
25         }
26         else if(result == 'V')
27         {
28             scoreV++;
29         }
30
31         if ((scoreK >= 11 || scoreV >= 11) && abs(scoreK - scoreV) >= 2)
32         {
33             if (scoreK > scoreV)
34             {
35                 gamescoreK++;
36             }
37             else
38             {
39                 gamescoreV++;
40             }
41
42             scoreK = 0;
43             scoreV = 0;
44         }
45     }
46
47     cout << gamescoreK << " " << gamescoreV << endl;
48     if (scoreK > 0 || scoreV > 0)
49     {
50         cout << scoreK << " " << scoreV << endl;
51     }
52
53     return 0;
54 }
```

```
15
VKKKKKKKKKKKVKKVV
1:0
0:3
```

Діаграма:





Зустрічі з командою

Зустрічались та обговорювали питання по епіку.



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.