

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  
Макович Маркіян

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, виконувати різні операції з ними (в мові C/C++)
- 2) Використання бібліотечних функцій

**Теоретичні відомості:**

1. Робота з файлами C
2. Робота з файлами C++

**Індивідуальний план опрацювання теорії:**

- Тема №1 Робота з файлами C (50 хв)  
(<https://www.youtube.com/watch?v=FeNqHytI0fA>)
- Тема №2 Робота з файлами C++ (40 хв)  
(<https://www.youtube.com/watch?v=SSNJ7alki-E>)

# Виконання роботи

## Завдання №1 VNS Lab 6 – 19 варіант

Знищити з рядка всі слова, які не є ідентифікаторами.

```
1 #include <iostream>
2 #include <string.h>
3 #include <vector>
4
5 using namespace std;
6
7 bool checkChar(int a)
8 {
9     for (int i = 34; i < 85; i++)
10     {
11         if (a == i)
12         {
13             return true;
14         }
15     }
16     return false;
17 }
18
19 int main()
20 {
21     char s[256] = {};
22     int n = 0;
23     vector<char> str;
24     int count = 0;
25     puts("Input your string:");
26     gets(s);
27
28     for (int i = 0; i < 256; i++)
29     {
30         if (checkChar(s[i]))
31         {
32             for (int j = i; j < 256; j++)
33             {
34                 if (s[j] != ' ')
35                 {
36                     str.push_back(s[j]);
37                     i++;
38                 }
39                 else
40                 {
41                     break;
42                 }
43             }
44             str.push_back(' ');
45         }
46     }
47
48     for (int i = 0; i < str.size(); i++)
49     {
50         cout << str[i];
51     }
52
53     return 0;
54 }
```

## Результати:

```
Input your string:
weqwpf fhdwpedh dwsh oiw oqiw oou10 99w 01w 091wd 092jsd02ij 02j osiojd aosj
weqwpf fhdwpedh dwsh oiw oqiw oou10 osiojd aosj
PS C:\Users\Маркіян\Desktop\epic_5> 
```

## Завдання №2 VNS Lab 8 – 19 варіант

Структура "Фільм":

- назва;
- режисер;
- країна;
- прибуток.

Знищити 2 елементи з кінця файлу, додати елемент після елемента із зазначеною назвою.

```
1  #include <iostream>
2  #include <stdio.h>
3  #include <unistd.h>
4
5  using namespace std;
6
7  struct films
8  {
9      char name[50];
10     char director[50];
11     char country[50];
12     long long profit;
13 };
14
15 void addData(const char *file)
16 {
17
18     films film[]{
19         {"Reservoir Dogs", "Quentin Tarantino", "USA", 2800000},
20         {"Shawshank Redemption", "Frank Darabont", "USA", 73000000},
21         {"12 Angry Men", "Sidney Arthur Lumet", "USA", 2000000},
22         {"Goodfellas", "Martin Scorsese", "USA", 47000000},
23         {"Seven", "David Andrew Leo Fincher", "USA", 327333559}};
24
25     FILE *filmFile = fopen(file, "wb");
26     if (filmFile == NULL)
27     {
28         cerr << "Fail to open a file";
29     }
30
31     for (int i = 0; i < 5; i++)
32     {
33         if (fwrite(&film[i], sizeof(films), 1, filmFile) != 1)
34         {
35             cerr << "An error occurred";
36             exit(0);
37         }
38     }
39     fclose(filmFile);
40 }
```

```

40 }
41
42 ~ void printData(const char *file)
43 {
44     films film;
45     FILE *filmFile;
46
47     filmFile = fopen(file, "rb");
48
49     if (filmFile == NULL)
50     {
51         cerr << "Fail to open a file";
52         exit(0);
53     }
54
55     while (fread(&film, sizeof(films), 1, filmFile) == 1)
56     {
57         cout << "Name: " << film.name << endl;
58         cout << "Director: " << film.director << endl;
59         cout << "Country: " << film.country << endl;
60         cout << "Profit: " << film.profit << "$" << endl;
61         cout << endl;
62     }
63     fclose(filmFile);
64 }
65
66 ~ void deleteData(const char *file)
67 {
68     FILE *filmFile;
69
70     filmFile = fopen(file, "rb+");
71
72     if (filmFile == NULL)
73     {
74         cerr << "Fail to open a file";
75         exit(0);
76     }
77
78     fseek(filmFile, 0, SEEK_END); // Переміщуємо вказівник на кінець файлу

```

```

88     long fileSize = ftell(filmFile); // Отримуємо розмір файлу
89     if (fileSize == 0)
90     {
91         cerr << "File is empty";
92         exit(0);
93     }
94     size_t filmSize = sizeof(films) * 2; // Розмір двох фільмів
95
96     long newSize = fileSize - filmSize;
97     if (ftruncate(fileno(filmFile), newSize) != 0) // Функція для обрізання файлу
98     {
99         cerr << "Error occurred" << endl;
100     }
101     else
102     {
103         cerr << "2 last elements were deleted" << endl;
104         cout << endl;
105     }
106     fclose(filmFile);
107
108     printData(file);
109 }
110
111 void addElement(const char *file, const char *filmName)
112 {
113     FILE *filmFile;
114     filmFile = fopen(file, "rb+");
115     FILE *newFilmFile;
116     newFilmFile = fopen("newFilmFile.dat", "wb");
117     if (filmFile == NULL)
118     {
119         cerr << "Fail to open a file";
120         exit(0);
121     }
122     if (newFilmFile == NULL)
123     {
124         cerr << "Fail to open a file";
125         exit(0);
126     }

```

```

118     }
119
120     films film;
121     films newFilm[]{
122         {"The Silence of the Lambs", "Jonathan Demme", "USA", 272742922}};
123
124     bool targetFound = false;
125     while (fread(&film, sizeof(films), 1, filmFile))
126     {
127         if (!targetFound && strcmp(film.name, fileName) == 0)
128         {
129             targetFound = true;
130             fwrite(&film, sizeof(films), 1, newFilmFile); // Записуємо поточний елемент
131             fwrite(&newFilm, sizeof(films), 1, newFilmFile); // Додаємо новий елемент
132         }
133         else
134         {
135             fwrite(&film, sizeof(films), 1, newFilmFile); // Продовжуємо копіювати інші елементи
136         }
137     }
138
139     fclose(filmFile);
140     fclose(newFilmFile);
141     cout << "With added film: " << endl;
142     cout << endl;
143     printData("newFilmFile.dat");
144 }
145
146 int main()
147 {
148     const char *fileName = "films.dat";
149     addData(fileName);
150     printData(fileName);
151     deleteData(fileName);
152     addElement(fileName, "Reservoir Dogs");
153     return 0;
154 }

```

## Результати:

```

Name: Reservoir Dogs
Director: Quentin Tarantino
Country: USA
Profit: 2800000$

Name: Shawshank Redemption
Director: Frank Darabont
Country: USA
Profit: 73000000$

Name: 12 Angry Men
Director: Sidney Arthur Lumet
Country: USA
Profit: 2000000$

Name: Goodfellas
Director: Martin Scorsese
Country: USA
Profit: 47000000$

Name: Seven
Director: David Andrew Leo Fincher
Country: USA
Profit: 327333559$

```

2 last elements were deleted

```

Name: Reservoir Dogs
Director: Quentin Tarantino
Country: USA
Profit: 2800000$

Name: Shawshank Redemption
Director: Frank Darabont
Country: USA
Profit: 73000000$

Name: 12 Angry Men
Director: Sidney Arthur Lumet
Country: USA
Profit: 2000000$

```

With added film:

```

Name: Reservoir Dogs
Director: Quentin Tarantino
Country: USA
Profit: 2800000$

Name: The Silence of the Lambs
Director: Jonathan Demme
Country: USA
Profit: 272742922$

Name: Shawshank Redemption
Director: Frank Darabont
Country: USA
Profit: 73000000$

Name: 12 Angry Men
Director: Sidney Arthur Lumet
Country: USA
Profit: 2000000$

```

### Завдання №3 VNS Lab 9 – 19 варіант

- 1) Скопіювати з файлу F1 у файл F2 всі рядки, у яких немає слів, що збігаються з першим словом.
- 2) Визначити кількість приголосних букв у першому рядку файлу F2.

```
1  #include <iostream>
2  #include <fstream>
3  #include <string>
4  #include <sstream>
5  using namespace std;
6
7  int main()
8  {
9      ifstream readFile("test.txt");
10     ofstream writeFile("out.txt", ios::in | ios::out | ios::trunc);
11
12     if (readFile.is_open() && writeFile.is_open())
13     {
14         string line;
15         while (getline(readFile, line))
16         {
17             istringstream iss(line);
18             string firstWord, checkWord, tempWord;
19             if (iss >> firstWord)
20                 tempWord = firstWord;
21
22             bool isInLine = false;
23             while (iss >> firstWord)
24             {
25                 if (firstWord == tempWord)
26                 {
27                     isInLine = true;
28                 }
29             }
30             if (!isInLine)
31             {
32                 writeFile << line << endl;
33             }
34         }
35     }
36 }
```



```

30         if (!isInLine)
31         {
32             writeFile << line << endl;
33         }
34     }
35 }
36 readFile.close();
37
38 writeFile.seekg(0, ios::beg);
39 string checkLine;
40 int count = 0;
41 while (getline(writeFile, checkLine))
42 {
43
44     for (char a : checkLine)
45     {
46         if (a == ' ')
47             continue;
48         if (a != 'a' || a != 'e' || a != 'u' || a != 'i' || a != 'y' || a != 'o')
49             count++;
50     }
51     break;
52 }
53
54 writeFile.seekp(0, ios::end); // Починаємо запис з кінця файлу
55 writeFile << "Num of consonant letters: " << to_string(count) << endl;
56 writeFile.close();
57
58 return 0;
59 }

```

## Результати:

test.txt: Блокнот

Файл Редагування Формат Вигляд Довідка

The dreamers are the saviors of world James Allen .  
 Life is about learning; when you stop learning, you die Tom Clancy .  
 Life is a dream for the wise, a game for the fool, a comedy for the rich, a tragedy for the poor Sholom Aleichem Life .  
 Our whole life is solving Is puzzles Erno Rubik .  
 Life is the art of drawing sufficient conclusions from insufficient premise Samuel Butler Life .  
 I will go anywhere as long as it's forward David Livingston .  
 When angry, count to four; when very angry, When swear Mark Twain .  
 I think being in love with life is a key to eternal youth Doug Hutchison .  
 Other things may change us, but we start and end with the family Anthony Brandt Other .  
 If I know what love is, it If is because of you Herman Hesse .

out.txt: Блокнот

Файл Редагування Формат Вигляд Довідка

Life is about learning; when you stop learning, you die Tom Clancy .  
 Our whole life is solving Is puzzles Erno Rubik .  
 I will go anywhere as long as it's forward David Livingston .  
 I think being in love with life is a key to eternal youth Doug Hutchison .  
 Num of consonant letters: 56



## Завдання №4 Algotester Lab 4v3

### Lab 4v3

Обмеження: 2 сек., 256 МБ

Вам дано масив, який складається з  $N$  додатних цілих чисел.

Ваше завдання - розділити його на три частини, по остачі від ділення на 3, по зростанню остачі (тобто спочатку йдуть числа, у яких остача 0, далі числа з остачею 1 і тоді паренті числа з остачею 2).

Далі необхідно ті елементи, остача від ділення на 3 яких парна посортувати по зростанню, а ті, у яких остача 1 - по спаданню.

Після цього видалити усі дублікати з масиву.

Виведіть результуючий масив.

#### Вхідні дані

У першому рядку  $N$  - кількість чисел.

У другому рядку  $N$  чисел  $a_i$  - елементи масиву.

```
1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4
5  using namespace std;
6
7  int main()
8  {
9      int N;
10     cin >> N;
11     vector<int> arr, arrSorted;
12     vector<int> divide0, divide1, divide2;
13
14     for (int i = 0; i < N; i++)
15     {
16         int a;
17         cin >> a;
18         arr.push_back(a);
19     }
20     sort(arr.begin(), arr.end());
21     auto last = unique(arr.begin(), arr.end());
22     arr.erase(last, arr.end());
23
24     for (int i = 0; i < arr.size(); i++)
25     {
26         if (arr[i] % 3 == 0)
27         {
28             divide0.push_back(arr[i]);
29         }
30         else if (arr[i] % 3 == 1)
31         {
32             divide1.push_back(arr[i]);
33         }
34         else if (arr[i] % 3 == 2)
35         {
36             divide2.push_back(arr[i]);
37         }
38     }
```

```

34     else if (arr[i] % 3 == 2)
35     {
36         divide2.push_back(arr[i]);
37     }
38 }
39
40 sort(divide0.begin(), divide0.end());
41 sort(divide2.begin(), divide2.end());
42 sort(divide1.begin(), divide1.end(), greater<int>());
43 for (int i = 0; i < divide0.size(); i++)
44 {
45     arrSorted.push_back(divide0[i]);
46 }
47
48 for (int i = 0; i < divide1.size(); i++)
49 {
50     arrSorted.push_back(divide1[i]);
51 }
52
53 for (int i = 0; i < divide2.size(); i++)
54 {
55     arrSorted.push_back(divide2[i]);
56 }
57
58 cout << arrSorted.size() << endl;
59 for (int i = 0; i < arrSorted.size(); i++)
60 {
61     cout << arrSorted[i] << " ";
62 }
63
64 return 0;
65 }

```

## Результати:

```

PS C:\Users\Маркіян> g++ 10
1 33 4 8 6 5 2 7 5 0
9
0 6 33 7 4 1 2 5 8
PS C:\Users\Маркіян\Desktop

```

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МБ)	Дії
11 днів тому	C++ 23	Зарховано	0.003	1.313	<a href="#">Перегляд</a>

## Завдання №4 Algotester Lab 4v2

### Lab 4v2

Обмеження: 1 сек., 256 MiB

Вам дано масив  $a$  з  $N$  цілих чисел.

Спочатку видаліть масиву  $a$  усі елементи що повторюються, наприклад масив  $[1, 3, 3, 4]$  має перетворитися у  $[1, 3, 4]$ .

Після цього оберніть посортовану версію масиву  $a$  на  $K$ , тобто при  $K = 3$  масив  $[1, 2, 3, 4, 5, 6, 7]$  перетвориться на  $[4, 5, 6, 7, 1, 2, 3]$ .

Виведіть результат.

#### Вхідні дані

У першому рядку цілі числа  $N$  та  $K$ .

У другому рядку  $N$  цілих чисел - елементи масиву  $a$ .

#### Вихідні дані

У першому рядку ціле число  $N$  - розмір множини  $a$ .

У наступному рядку  $N$  цілих чисел - множина  $a$ .

#### Обмеження

$$1 \leq N, K \leq 1000$$

$$0 \leq a_i \leq 100$$

```
1 #include <iostream>
2 #include <vector>
3
4 using namespace std;
5
6 void merge(vector<int> &arr, int low, int mid, int high)
7 {
8     int n1 = mid - low + 1;
9     int n2 = high - mid;
10
11     vector<int> leftArray(n1);
12     vector<int> rightArray(n2);
13
14     for (int i = 0; i < n1; ++i)
15         leftArray[i] = arr[low + i];
16     for (int j = 0; j < n2; ++j)
17         rightArray[j] = arr[mid + 1 + j];
18
19     int i = 0, j = 0, k = low;
20     while (i < n1 && j < n2)
21     {
22         if (leftArray[i] <= rightArray[j])
23         {
24             arr[k] = leftArray[i];
25             ++i;
26         }
27         else
28         {
29             arr[k] = rightArray[j];
30             ++j;
31         }
32         ++k;
33     }
34
35     while (i < n1)
36     {
37         arr[k] = leftArray[i];
38         ++i;
39         ++k;
40     }
41
42     while (j < n2)
43     {
44         arr[k] = rightArray[j];
45         ++j;
46         ++k;
47     }
48 }
49
50 void mergeSort(vector<int> &arr, int low, int high)
51 {
52     if (low < high)
53     {
54         int mid = (low + high) / 2;
55         mergeSort(arr, low, mid);
56         mergeSort(arr, mid + 1, high);
57         merge(arr, low, mid, high);
58     }
59     return;
60 }
61
62 int main()
63 {
```

```

63 int main()
64 {
65     int N, K;
66     cin >> N >> K;
67     vector<int> arr;
68     for (int i = 0; i < N; i++)
69     {
70         int a;
71         cin >> a;
72         arr.push_back(a);
73     }
74     int low = 0;
75     int high = arr.size() - 1;
76
77     mergeSort(arr, low, high);
78     vector<int> uniqueArr;
79     for (int i = 0; i < N; i++)
80     {
81         if (arr[i] != arr[i + 1])
82         {
83             uniqueArr.push_back(arr[i]);
84         }
85     }
86
87     for (int i = 0; i < K; i++)
88     {
89         uniqueArr.push_back(uniqueArr[0]);
90         uniqueArr.erase(uniqueArr.begin());
91     }
92     cout << uniqueArr.size() << endl;
93     for (int i = 0; i < uniqueArr.size(); i++)
94     {
95         if (i == uniqueArr.size() - 1)
96         {
97             cout << uniqueArr[i];
98             break;
99         }
100         cout << uniqueArr[i] << " ";
101     }

```

## Результати:

```

ych }
10 11
5 6 2 3 1 2 3 3 4 7
7
5 6 7 1 2 3 4
PS C:\Users\Маркіян\Desktop\epic

```

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МБ)	Дії
11 днів тому	C++ 23	Зарховано	0.003	1.207	<a href="#">Перегляд</a>

## Завдання №5 Algotester Lab 6v2

### Lab 6v2

Обмеження: 2 сек., 256 МБ

У вас є шахова дошка розміром  $8 \times 8$  та дуже багато фігур.

Кожна клітинка може мати таке значення:

- Пуста клітинка  $O$
- Пішак  $P$
- Тур  $R$
- Кінь  $N$
- Слон  $B$
- Король  $K$
- Королева  $Q$

Вам дають позиції фігур на дошці (всі фігури одного кольору, кількість кроків може бути  $> 1$ ).

Далі йдуть  $Q$  запитів з координатами клітинки  $(x, y)$ . На кожне запит ви маєте вивести стрілку  $s_i$  - посортовані за алфавітом букви фігур, які атакують цю клітинку (пішак атакує вніж).

У випадку, якщо на клітинці стоїть якась фігура - виведіть символ  $X$ .

```
1 #include <iostream>
2 #include <vector>
3 #include <set>
4 #include <algorithm>
5
6 using namespace std;
7
8 bool isWithinBounds(int row, int col)
9 {
10     return row >= 0 && row < 8 && col >= 0 && col < 8;
11 }
12
13 void processPawnAttacks(vector<set<char>> &attack, int row, int col, char piece)
14 {
15     if (isWithinBounds(row + 1, col - 1))
16         attack[(row + 1) * 8 + (col - 1)].insert(piece);
17     if (isWithinBounds(row + 1, col + 1))
18         attack[(row + 1) * 8 + (col + 1)].insert(piece);
19 }
20
21 void processRookAttacks(vector<set<char>> &attack, int row, int col, char piece)
22 {
23     for (int i = 0; i < 8; ++i)
24     {
25         if (i != col)
26             attack[row * 8 + i].insert(piece);
27         if (i != row)
28             attack[i * 8 + col].insert(piece);
29     }
30 }
31
32 void processBishopAttacks(vector<set<char>> &attack, int row, int col, char piece)
33 {
34     for (int offset = 1; offset < 8; ++offset)
35     {
36         if (isWithinBounds(row + offset, col + offset))
37             attack[(row + offset) * 8 + (col + offset)].insert(piece);
38         if (isWithinBounds(row + offset, col - offset))
39             attack[(row + offset) * 8 + (col - offset)].insert(piece);
40         if (isWithinBounds(row - offset, col + offset))
41             attack[(row - offset) * 8 + (col + offset)].insert(piece);
42         if (isWithinBounds(row - offset, col - offset))
43             attack[(row - offset) * 8 + (col - offset)].insert(piece);
44     }
45 }
```

```

81         attack[(row + offset) * 8 + (col + offset)].insert(piece);
82     if (!isWithinBounds(row + offset, col + offset))
83         attack[(row - offset) * 8 + (col - offset)].insert(piece);
84     }
85 }
86
87 void processKnightAttacks(vector<set<char>> &attack, int row, int col, char piece)
88 {
89     const int moves[8][2] = {{-2, -1}, {-2, 1}, {2, -1}, {2, 1}, {-1, -2}, {1, -2}, {-1, 2}, {1, 2}};
90     for (const auto &move : moves)
91     {
92         int newRow = row + move[0], newCol = col + move[1];
93         if (isWithinBounds(newRow, newCol))
94             attack[newRow * 8 + newCol].insert(piece);
95     }
96 }
97
98 void processKingAttacks(vector<set<char>> &attack, int row, int col, char piece)
99 {
100     for (int dr = -1; dr <= 1; ++dr)
101     {
102         for (int dc = -1; dc <= 1; ++dc)
103         {
104             if (dr != 0 || dc != 0)
105             {
106                 int newRow = row + dr, newCol = col + dc;
107                 if (isWithinBounds(newRow, newCol))
108                     attack[newRow * 8 + newCol].insert(piece);
109             }
110         }
111     }
112 }
113
114 int main()
115 {
116     vector<string> chessboard(8);
117     for (int i = 0; i < 8; ++i)
118     {
119         cin >> chessboard[i];
120     }

```

```

121
122 int main()
123 {
124     vector<string> chessboard(8);
125     for (int i = 0; i < 8; ++i)
126     {
127         cin >> chessboard[i];
128     }
129
130     vector<set<char>> attack(64);
131
132     for (int row = 0; row < 8; ++row)
133     {
134         for (int col = 0; col < 8; ++col)
135         {
136             char piece = chessboard[row][col];
137             if (piece == 'P')
138             {
139                 processPawnAttacks(attack, row, col, 'P');
140             }
141             else if (piece == 'R')
142             {
143                 processRookAttacks(attack, row, col, 'R');
144             }
145             else if (piece == 'B')
146             {
147                 processBishopAttacks(attack, row, col, 'B');
148             }
149             else if (piece == 'N')
150             {
151                 processKnightAttacks(attack, row, col, 'N');
152             }
153             else if (piece == 'K')
154             {
155                 processKingAttacks(attack, row, col, 'K');
156             }
157             else if (piece == 'Q')
158             {
159                 processRookAttacks(attack, row, col, 'Q');
160                 processBishopAttacks(attack, row, col, 'Q');
161             }
162         }
163     }

```

```

114     }
115 }
116
117 int queries;
118 cin >> queries;
119
120 while (queries-- > 0)
121 {
122     int queryRow, queryCol;
123     cin >> queryRow >> queryCol;
124     --queryRow;
125     --queryCol;
126
127     if (chessboard[queryRow][queryCol] != '0')
128     {
129         cout << "X\n";
130     }
131     else
132     {
133         const auto &attackingPieces = attack[queryRow * 8 + queryCol];
134         if (attackingPieces.empty())
135         {
136             cout << "0\n";
137         }
138         else
139         {
140             for (char piece : attackingPieces)
141             {
142                 cout << piece;
143             }
144             cout << endl;
145         }
146     }
147 }
148
149 return 0;
150 }

```

## Результати:

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МБ)	Дії
3 години тому	C++ 23	Зараховано	0.003	1.234	<a href="#">Перегляд</a>



## Завдання №6 Class Practice Work

```
1 #include <iostream>
2 #include <string>
3 #include <fstream>
4
5 using namespace std;
6
7 enum FileOpResult
8 {
9     Success,
10    Failure
11 };
12
13 FileOpResult write_to_file(string name, string content)
14 {
15
16     if (name.empty() || content.empty())
17     {
18         return Failure;
19     }
20
21     ofstream file(name, ios::out | ios::trunc);
22
23     if (!file.is_open())
24     {
25         return Failure;
26     }
27
28     file << content;
29
30     if (!file.is_open())
31     {
32         file.close();
33         return Failure;
34     }
35     file.close();
36     return Success;
37 }
38
39 FileOpResult copy_file(string fileFrom, string fileTo)
40 {
41     if (fileFrom.empty() || fileTo.empty())
```

```
42     {
43         return Failure;
44     }
45     ifstream src(fileFrom);
46     if (!src.is_open())
47     {
48         return Failure;
49     }
50
51     ofstream destination(fileTo, ios::out | ios::trunc);
52     if (!destination.is_open())
53     {
54         src.close();
55         return Failure;
56     }
57
58     destination << src.rdbuf();
59
60     if (!destination)
61     {
62         src.close();
63         destination.close();
64         return Failure;
65     }
66
67     src.close();
68     destination.close();
69
70     if (!src || !destination)
71     {
72         return Failure;
73     }
74
75     return Success;
76 }
77
78 int main()
79 {
```

```

78  int main()
79  {
80      string content, name, fileFrom, fileTo;
81      cout << "File name:" << endl;
82      getline(cin, name);
83      cout << "What do you want to write?" << endl;
84      getline(cin, content);
85
86      FileOpResult result = write_to_file(name, content);
87
88      if (result == Success)
89      {
90          cout << "Everything is correct" << endl;
91      }
92      else
93      {
94
95          cout << "Something went wrong" << endl;
96      }
97      cout << endl;
98      cout << "From which file you want to copy?" << endl;
99      getline(cin, fileFrom);
100     cout << "In which file you want to write?" << endl;
101     getline(cin, fileTo);
102     FileOpResult result1 = copy_file(fileFrom, fileTo);
103     if (result1 == Success)
104     {
105         cout << "Everything is correct" << endl;
106     }
107     else
108     {
109
110         cout << "Something went wrong" << endl;
111     }
112
113     return 0;
114 }

```

## Результати:

```

ian_makovych }
File name:
file.txt
What do you want to write?
Hello my friend!
Everything is correct

From which file you want to copy?
file.txt
In which file you want to write?
fileCopy.txt
Everything is correct
PS C:\Users\Маркіян\Desktop\epic_5>

```

fileCopy.txt: Блокнот

Файл Редагування Формат Вигляд Довідка

Hello my friend!

file.txt: Блокнот

Файл Редагування Формат Вигляд Довідка

Hello my friend!

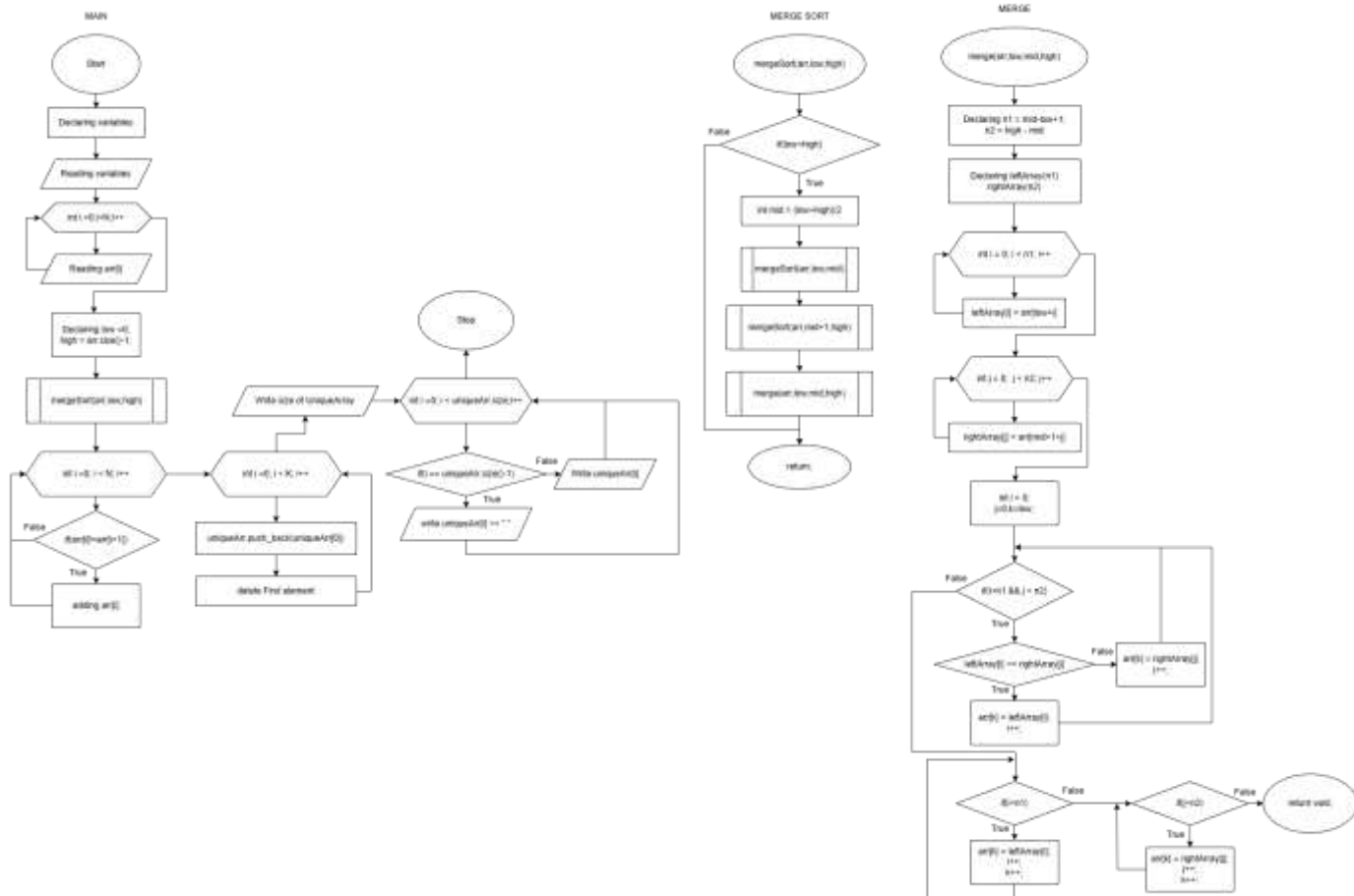
## Завдання №7 Self Practice Work

```
1  #include <iostream>
2  #include <string>
3  #include <vector>
4
5  using namespace std;
6
7  int main()
8  {
9      int rows, columns;
10     cin >> rows >> columns;
11
12     vector<string> crossWord(rows);
13     vector<vector<int>> toDelete(rows, vector<int>(columns, 0));
14
15     for (int i = 0; i < rows; i++)
16     {
17         cin >> crossWord[i];
18     }
19
20     for (int i = 0; i < rows; i++)
21     {
22         for (int j = 0; j < columns; j++)
23         {
24             char a = crossWord[i][j];
25
26             for (int k = 0; k < rows; k++)
27             {
28                 if (i != k && crossWord[k][j] == a)
29                 {
30                     toDelete[k][j] = 1;
31                     toDelete[i][j] = 1;
32                 }
33             }
34
35             for (int k = 0; k < columns; k++)
36             {
37                 if (j != k && crossWord[i][k] == a)
38                 {
39                     toDelete[i][k] = 1;
40                     toDelete[i][j] = 1;
41                 }
42             }
43         }
44     }
45
46     for (int i = 0; i < rows; i++)
47     {
48         for (int j = 0; j < columns; j++)
49         {
50             if (toDelete[i][j] == 0)
51             {
52                 cout << crossWord[i][j];
53             }
54         }
55     }
56
57     return 0;
58 }
59
```

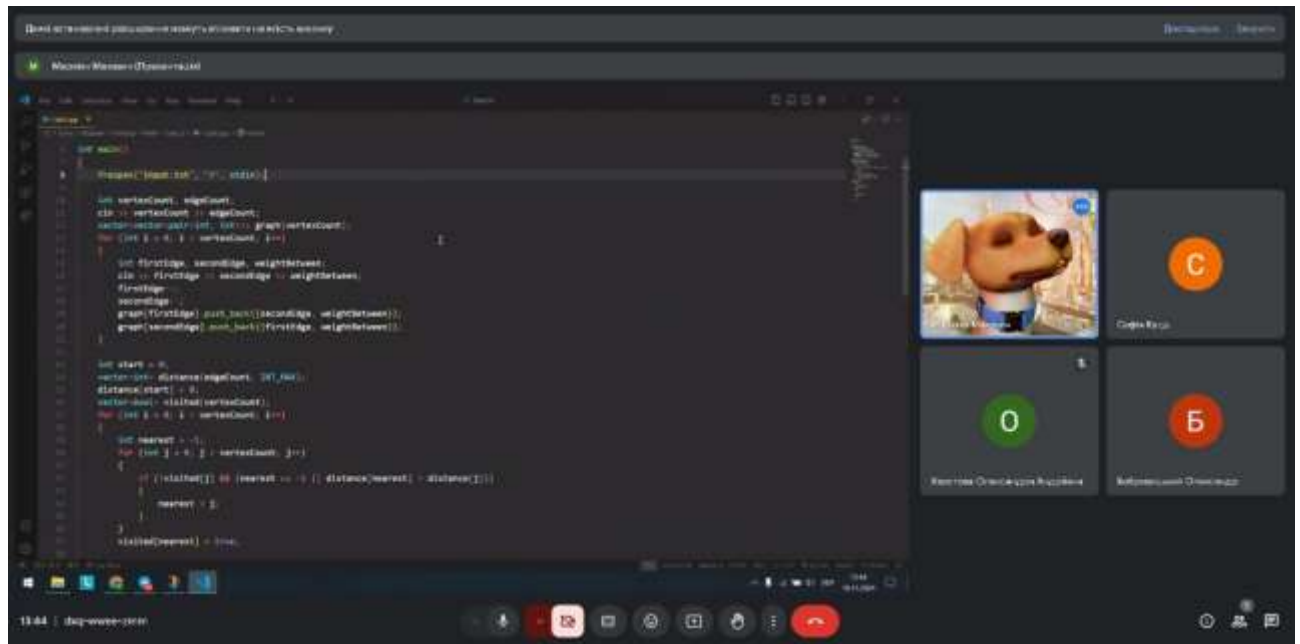
### Результати:

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МБ)	Дії
Година тому	C++ 23	Зарховано	0.005	1.090	<a href="#">Перегляд</a>

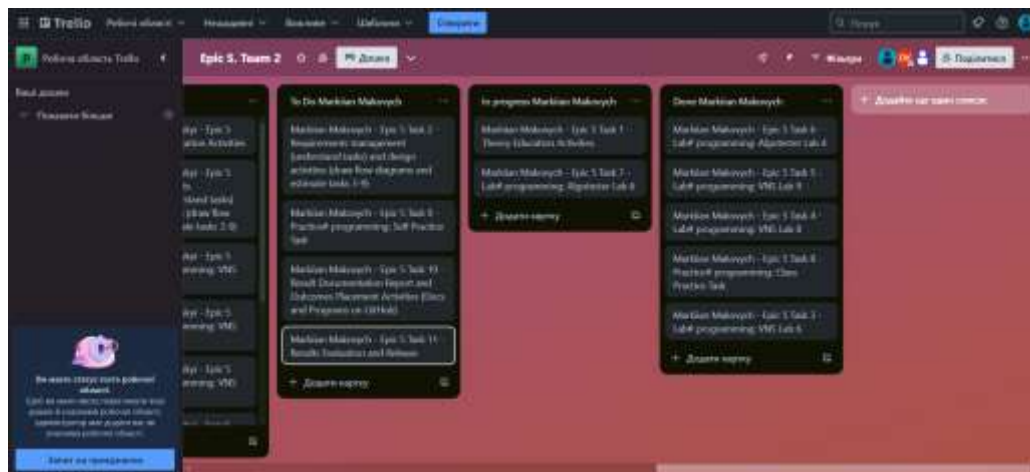
## Діаграма для Algotester Lab 4v2



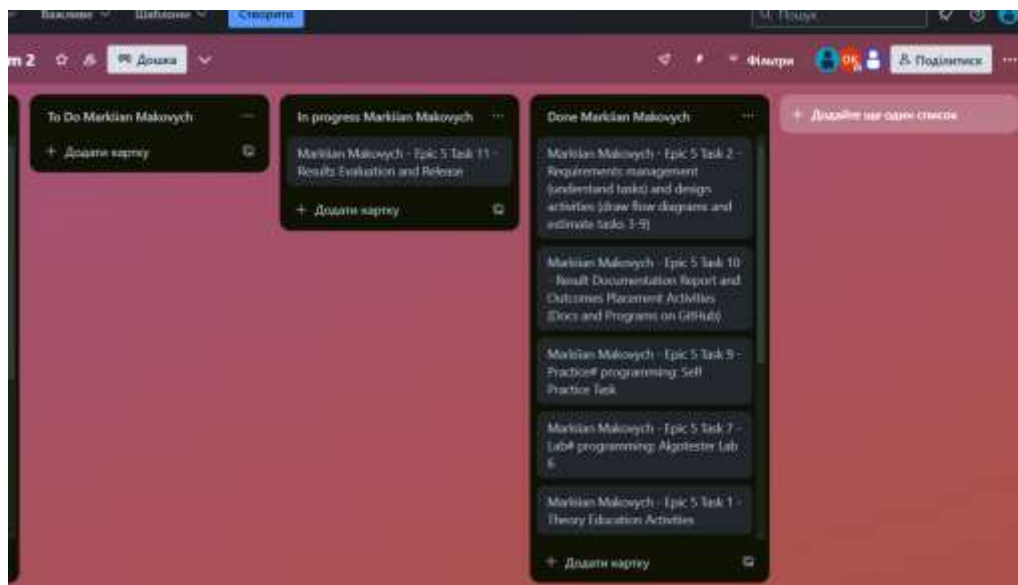
## Робота в команді:



## Прогрес Trello(1):



## Прогрес Trello(2):



## **Висновок:**

Під час виконання лабораторної роботи, я навчився використовувати різні методи для роботи з файлами, вивчив кілька бібліотечних функцій для швидшої роботи з масивами