

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 ІІІ-12

Бобровицький Олександр Сергійович

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: ознайомитись з темами лабораторної, опрацювати їх теоретично та навчитися використовувати отриманні знання для вирішення практичних задач

Теоретичні відомості:

Тема №1 : Робота з файлами:

- Джерела:
 - <https://www.geeksforgeeks.org/file-handling-c-classes/>
 - https://www.w3schools.com/cpp/cpp_files.asp
- Що опрацьовано:
 - <https://www.geeksforgeeks.org/file-handling-c-classes/>
 - https://www.w3schools.com/cpp/cpp_files.asp
- Статус: Ознайомлений
- Початок опрацювання теми: 15.10
- Звершення опрацювання теми: 15.11

Тема №1 : Робота з бібліотеками:

- Джерела:
 - <https://cplusplus.com/reference/>
 - <https://www.geeksforgeeks.org/how-do-i-create-a-library-in-cpp/>
 - <https://stackoverflow.com/questions/140061/when-to-use-dynamic-vs-static-libraries>
- Що опрацьовано:
 - <https://cplusplus.com/reference/>
 - <https://www.geeksforgeeks.org/how-do-i-create-a-library-in-cpp/>
 - <https://stackoverflow.com/questions/140061/when-to-use-dynamic-vs-static-libraries>
- Статус: Ознайомлений
- Початок опрацювання теми: 01.09
- Звершення опрацювання теми: 10.11

Виконання роботи:

1. Опрацювання завдання та вимог до програм та середовища:

Завдання №1 VNS Lab 6

- 11 варіант
- Деталі завдання:
 - Задано рядок, що складається із символів. Символи поєднуються в слова. Слова одне від одного відокремлюються одним або декількома пробілами. Наприкінці тексту ставиться крапка. Текст містить не більше 255 символів. Виконати ввід рядка, використовуючи функцію gets(s) і здійснити обробку рядка у відповідності зі своїм варіантом.
 - Перетворити рядок таким чином, щоб всі слова в ньому були надруковані навпаки.

Завдання №1 VNS Lab 8

- 11 варіант
- Деталі завдання:

- Сформувати двійковий файл із елементів, заданої у варіанті структури, роздрукувати його вміст, 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 "Відеокасета":
 - назва фільму;
 - режисер;
 - тривалість;
 - ціна.
- Знищити всі елементи із ціною вищою заданої, додати 3 елементи в кінець файлу.

Завдання №1 VNS Lab 9

- 11 варіант
- Деталі завдання:
 - Створити текстовий файл F1 не менше, ніж з 10 рядків і записати в нього інформацію.
 - Скопіювати з файлу F1 у файл F2 всі рядки, крім того рядка, що містить найкоротше слово.
 - Надрукувати номер цього рядка.

Завдання №4, 5 Algotester Lab 4

- 1 варіант
- Деталі завдання:

Вам дано 2 цілих чисел масиви, розміром N та M .

Ваше завдання вивести:

 1. Різницю $N-M$
 2. Різницю $M-N$
 3. Їх перетин
 4. Їх об'єднання
 5. Їх симетричну різницю

Input

У першому рядку ціле число N - розмір масиву 1

У другому рядку N цілих чисел - елементи масиву 1

У третьому рядку ціле число M - розмір масиву 2

У четвертому рядку M цілих чисел - елементи масиву 2

Output

Вивести результат виконання 5 вищезазначених операцій у форматі:

У першому рядку ціле число N - розмір множини

У наступному рядку N цілих чисел - посортована у порядку зростання множина

Constraints

$$1 \leq N, M \leq 100$$

$$1 \leq n_i, m_i \leq 100$$

- Важливі деталі для врахування в імплементації програми

Пам'ятайте, ви маєте написати 2 варіанти розв'язку, один з використанням засобів STL (`std::set_intersection`, `std::set_symmetric_difference`, `std::set_difference`, `std::set_union`), інший зі своєю реалізацією. Своє сортування можна не писати.

Завдання №6 Algotester Lab 6

- 3 варіант
- Деталі завдання:

У Клінта в черговий раз виключилось світло і йому немає чим зайнятися. Так як навіть це не заставить його подивитися збережені відео про програмування на ютубі - він вирішив придумати свою гру на основі sudoku.

Гра виглядає так

Є поле розміром $N \times N$, в якому частина клітинок заповнена цифрами, а частина клітинок пуста (позначаються нулем). Також у нього є Q пар координат X та Y .

Завданням гри є написати до кожної координати скільки чисел туди можна вписати (якщо вона пуста) і які це числа (обов'язково в посортовані по зростанню!). В клітинку можна вписати лише ті числа, які не зустрічаються в рядку та стовбці, які перетинаються у цій клітинці.

Під час гри поле не міняється!

Також необов'язково, щоб це було валідне sudoku! Якщо є клітинка, в яку не можна вписати ніяку цифру - виведіть 0.

Також допускаються рядки та стовпці, в яких цифра записана кілька разів.

Input

У першому рядку ціле число N - розмір поля для гри

У N наступних рядках стрічка row_i яка складається з N цифр - i -й рядок.

Ціле число Q - кількість запитань

У наступних Q рядках 2 цілих числа x_j, y_j - координати клітинок j -го запитання

Output

Q разів відповідь у наступному форматі:

Натуральне число M - кількість цифр, які можна вписати в клітинку

M цифр розділених пробілом - можливі цифри

- Важливі деталі для врахування в імплементації програми:

Constraints

$$1 \leq N \leq 9$$

$$|row_i| = N$$

$$row_i \in 1..9$$

$$1 \leq Q \leq 1000$$

$$1 \leq x, y \leq N$$

Завдання №7 Class Practice Work

- 1 варіант
- Деталі завдання:
 - Частина 1

Реалізувати функцію створення файлу і запису в нього даних:

```
enum FileOpResult { Success, Failure, ... };
```

```
FileOpResult write_to_file(char *name, char *content);
```

Умови задачі:

- створити файл із заданим ім'ям; якщо файл існує – перезаписати його вміст
- написати код стійкий до різних варіантів вхідних параметрів
- name – ім'я, може не включати шлях
- записати у файл вміст стрічки content, прочитати content із стандартного вводу
- повернути статус операції: Success – все пройшло успішно, Failure – файл не вдалося створити, або збій операції відкриття, запису даних, чи закриття файлу.

- Частина 2

```
enum FileOpResult { Success, Failure, ... };
```

```
FileOpResult copy_file(char *file_from, char *file_to);
```

Умови задачі:

- копіювати вміст файлу з ім'ям file_from у файл з ім'ям file_to; написати код стійкий до різних варіантів вхідних параметрів, обробити всі можливі варіанти відсутності одного з файлів
- file_from, file_to – можуть бути повним або відносним шляхом
- повернути статус операції: Success – все пройшло успішно, Failure – файл не вдалося створити, або збій операції відкриття, читання чи запису даних, закриття файлу.

Завдання №8 Self-Practice Work

- 1 варіант
- Деталі завдання:

Your task is to guess the number n between 1 and 10^9 , inclusive.

To make a guess, print a line with a number that should be an integer between 1 and 10^9 , inclusive. After that, read a response line containing one of the following characters:

- < means your guess is less than n ;
- > means your guess is greater than n ;
- = means your guess is correct, please exit your solution.

You can make up to 47 guesses. Note that if a guess is invalid, your solution will immediately get the **Wrong answer** verdict.

Constraints

$$1 \leq n \leq 10^9.$$

Notes

If $n = 47$, the input and output might look like this:

2. Дизайн та планована оцінка часу виконання завдань:

Програма №1 VNS Lab 6

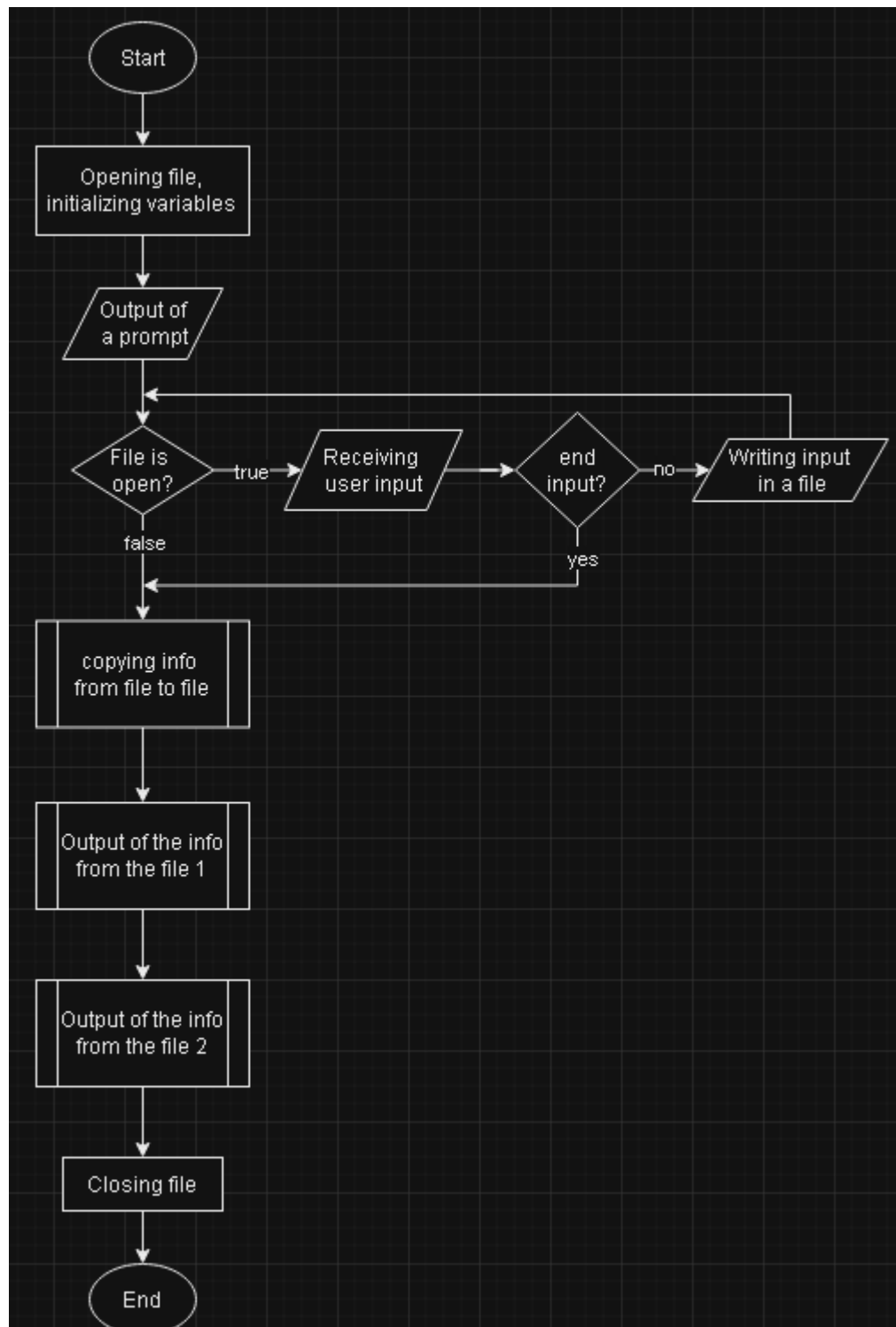
Планований час на реалізацію – 20min

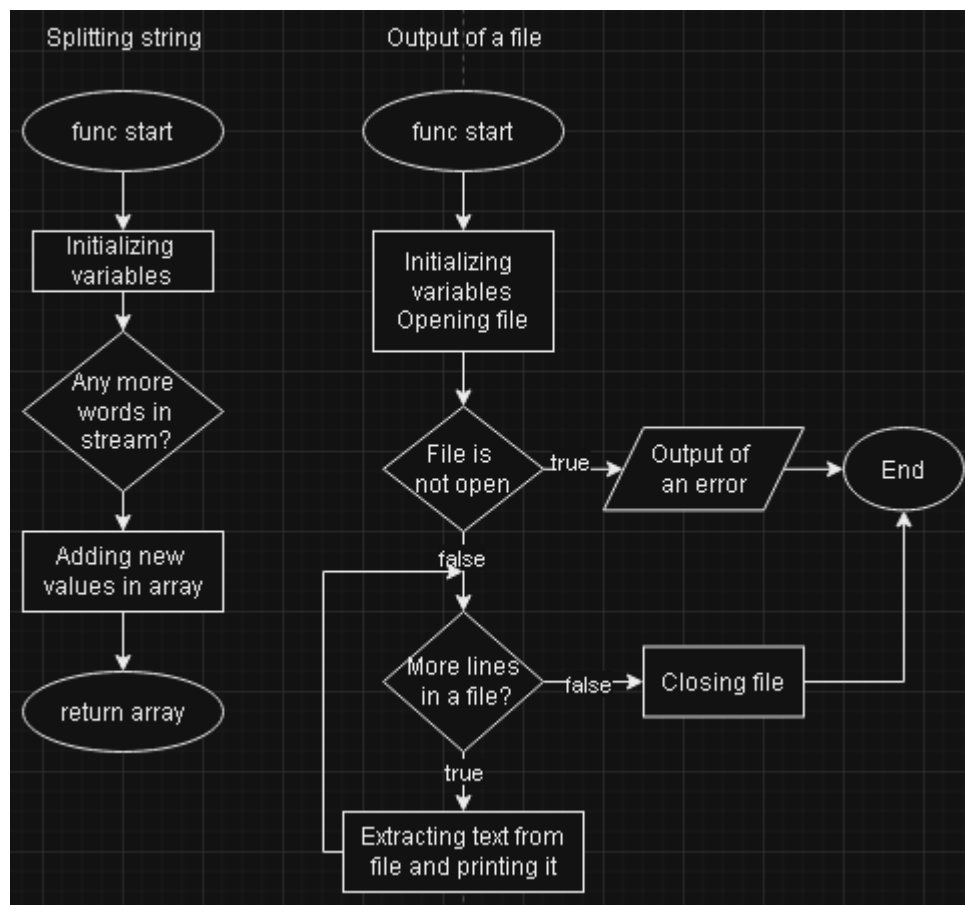
Програма №2 VNS Lab 8

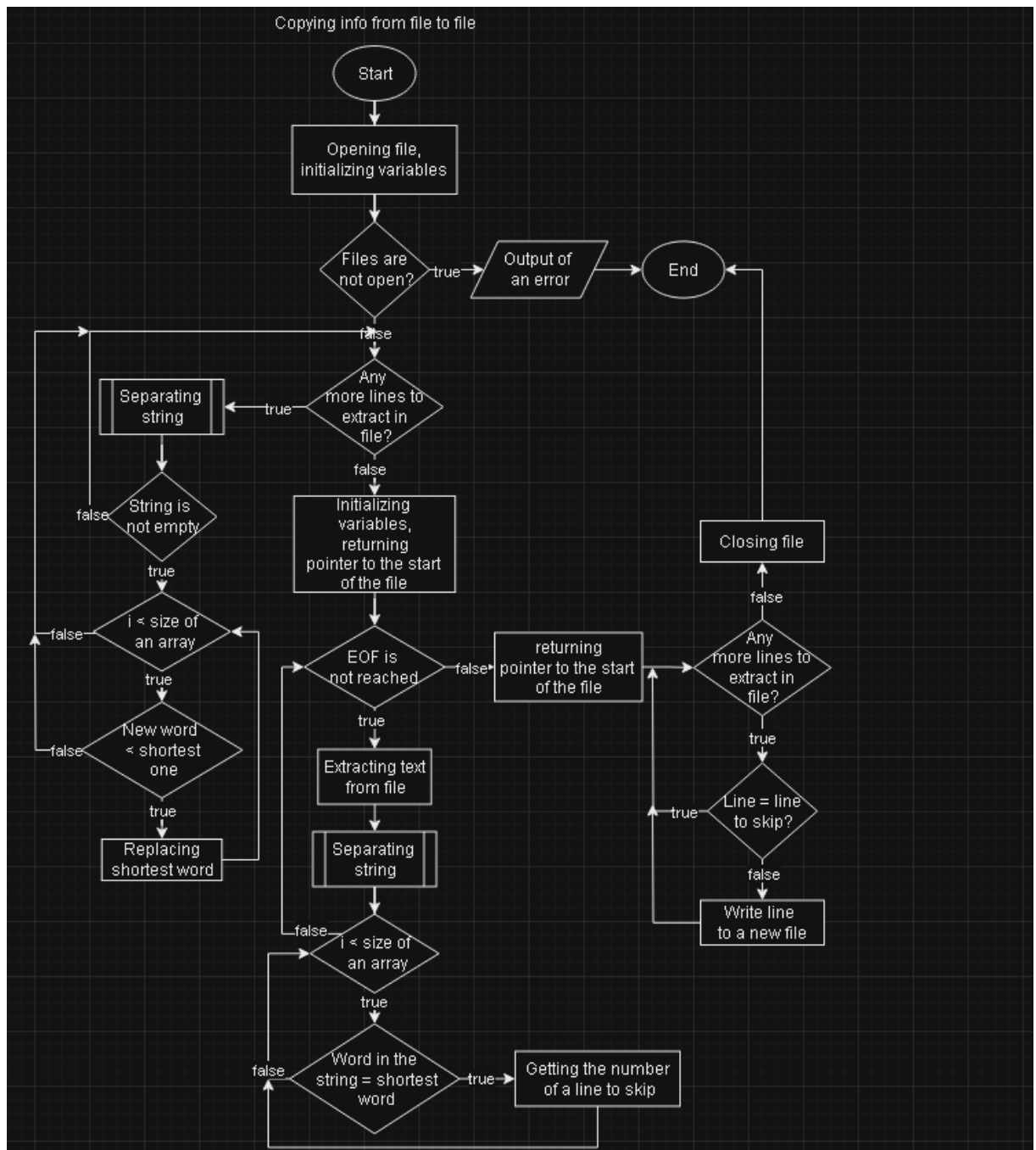
Планований час на реалізацію - 40min

Програма №3 VNS Lab 9

– Блок-схема







Планований час на реалізацію – 40min

Програма №4 Algotester Lab 4.1

Планований час на реалізацію – 20min

Програма №5 Algotester Lab 4.2

Планований час на реалізацію – 30min

Програма №6 Algotester Lab 6

Планований час на реалізацію – 40min

Програма №7 Class Practice Work

Планований час на реалізацію – 30min

Програма №8 Self Practice Work

Планований час на реалізацію – 20min

3. Код програм з посиланням на зовнішні ресурси:

Завдання №1

Посилання на файл програми у пул-запиті GitHub

A screenshot of a code editor with a dark background and light-colored text. The code is a C++ program that reads a line of text, splits it into words, reverses each word, and prints the result. The code is numbered from 1 to 28. The editor has three colored window control buttons (red, yellow, green) in the top left corner.

```
1  #include <bits/stdc++.h>
2
3  using namespace std;
4
5  int main()
6  {
7      char s[256];
8      gets(s);
9
10     char *token = strtok(s, " ");
11
12     vector<string> SeparatedWords;
13
14     while (token != nullptr)
15     {
16         SeparatedWords.push_back(token);
17
18         token = strtok(nullptr, " ");
19     }
20
21     for(int i = 0; i < SeparatedWords.size(); i++)
22     {
23         reverse(SeparatedWords[i].begin(), SeparatedWords[i].end());
24         cout << SeparatedWords[i] << " ";
25     }
26
27     return 0;
28 }
```

Код до завдання №1

Завдання №2

Посилання на файл програми у пул-запиті GitHub

```

1  #include <bits/stdc++.h>
2
3  using namespace std;
4
5  struct Videotape{
6      string name;
7      string director;
8      double RunningTime;
9      double price;
10
11  void input()
12  {
13      cout << "Enter the new name of the videotape" << endl;
14      getline(cin >> ws, name);
15
16      cout << "Enter the new name of the director" << endl;
17      getline(cin, director);
18
19      cout << "Enter the new running time of the film" << endl;
20      cin >> RunningTime;
21
22      cout << "Enter the new price of the videotape" << endl;
23      cin >> price;
24  }
25  };
26
27
28 void WriteToFile(string &fileName, vector<Videotape> &tapes)
29 {
30     ofstream fin;
31
32     fin.open(fileName, ios::binary);
33
34     if(!fin)
35     {
36         cerr << "Error: unable to open the file";
37         return;
38     }
39
40     for(int i = 0; i < tapes.size(); i++)
41     {
42         fin << tapes[i].name << endl << tapes[i].director << endl << tapes[i].RunningTime << endl << tapes[i].price << endl << endl;
43     }
44
45     fin.close();
46 }
47
48 void DeleteByPrice(string &fileName, vector<Videotape> &tapes, double MaxPrice)
49 {
50     ofstream fdel;
51
52     fdel.open(fileName, ios::binary | ios::trunc);
53
54     if(!fdel)
55     {
56         cerr << "Error: unable to open the file";
57         return;
58     }
59
60     for(int j = 0; j < tapes.size(); j++)
61     {
62         if(tapes[j].price > MaxPrice)
63         {
64             tapes.erase(tapes.begin() + j);
65         }
66     }
67
68     for(int i = 0; i < tapes.size(); i++)
69     {
70         fdel << tapes[i].name << endl << tapes[i].director << endl << tapes[i].RunningTime << endl << tapes[i].price << endl << endl;
71     }
72
73     fdel.close();
74 }
75
76 void OutputToFile(string &fileName)
77 {
78     string line;
79     ifstream fout;
80
81     fout.open(fileName, ios::binary);
82
83     if(!fout)
84     {
85         cerr << "Error: unable to open the file";
86         return;
87     }
88
89     while(fout)
90     {
91         getline(fout, line);
92         cout << line << endl;
93     }
94
95     fout.close();
96 }
97
98 void AddToInd(string &fileName, vector<Videotape> &tapes)
99 {
100     ofstream fin;
101
102     fin.open(fileName, ios::binary | ios::app);
103
104     if(!fin)
105     {
106         cerr << "Error: unable to open the file";
107         return;
108     }
109
110     for(int i = 0; i < tapes.size(); i++)
111     {
112         fin << endl << tapes[i].name << endl << tapes[i].director << endl << tapes[i].RunningTime << endl << tapes[i].price << endl;
113     }
114
115     fin.close();
116 }
117
118
119 int main()
120 {
121     string fileName = "VideotapeList.bin";
122     double MaxPrice = 0;
123
124     vector<Videotape> InitialTapes = {
125         {"film1", "Director1", 150.0, 15.0},
126         {"film2", "Director2", 125.0, 20.0},
127         {"film3", "Director3", 90.0, 10.0},
128         {"film4", "Director4", 100.0, 25.0}
129     };
130
131     cout << "Enter the max price:" << endl;
132     cin >> MaxPrice;
133
134     WriteToFile(fileName, InitialTapes);
135
136     OutputToFile(fileName);
137
138     DeleteByPrice(fileName, InitialTapes, MaxPrice);
139
140     vector<Videotape> NewTapes(3);
141     cout << "Add 3 more videotapes " << endl;
142     for(int i = 0; i < 3; i++)
143     {
144         NewTapes[i].input();
145     }
146
147     AddToInd(fileName, NewTapes);
148
149     cout << "New list of videotapes: " << endl;
150     OutputToFile(fileName);
151
152     return 0;
153 }

```

Код до завдання №2

Завдання №3

Посилання на файл програми у пул-запиті GitHub

```

1 #include <bits/stdc++.h>
2
3 using namespace std;
4
5 vector<string> StringsPliter(string &line)
6 {
7     stringstream s(line);
8     string word;
9     vector<string> SeparatedString;
10
11     while (s >> word) {
12         SeparatedString.push_back(word);
13     }
14
15     return SeparatedString;
16 }
17
18 void OutputOfAFile(string &FileName)
19 {
20     string line;
21     ifstream fin;
22
23     fin.open(FileName);
24
25     if(!fin)
26     {
27         cerr << "Error: unable to open the file";
28         return;
29     }
30
31     while(fin)
32     {
33         getline(fin, line);
34
35         cout << line << endl;
36     }
37
38     fin.close();
39 }
40
41 void CopyFileInFo(string &FileName1, string &FileName2)
42 {
43     string line;
44     ifstream fin;
45     ofstream fout;
46     fin.open(FileName1);
47     fout.open(FileName2);
48
49     vector<string> SeparatedString;
50     string ShortestWord;
51
52     if (!fin || !fout) {
53         cerr << "Error opening files" << endl;
54         return;
55     }
56
57     while(getline(fin, line))
58     {
59         SeparatedString = StringsPliter(line);
60
61         if(!SeparatedString.empty())
62         {
63             ShortestWord = SeparatedString[0];
64
65             for(int i = 0; i < SeparatedString.size(); i++)
66             {
67                 if(SeparatedString[i].length() < ShortestWord.length())
68                 {
69                     ShortestWord = SeparatedString[i];
70                 }
71             }
72         }
73     }
74
75     int count = 0;
76     int SkippedLine = -1;
77
78     fin.clear();
79     fin.seek(0, ios::beg);
80
81     while(fin)
82     {
83         getline(fin, line);
84         SeparatedString = StringsPliter(line);
85         count++;
86
87         for(int i = 0; i < SeparatedString.size(); i++)
88         {
89             if(SeparatedString[i] == ShortestWord)
90             {
91                 SkippedLine = count;
92             }
93         }
94     }
95
96     count = 0;
97
98     fin.clear();
99     fin.seek(0, ios::beg);
100
101     while(getline(fin, line))
102     {
103         count++;
104         if(count != SkippedLine)
105         {
106             fout << line << endl;
107         }
108     }
109
110     fin.close();
111     fout.close();
112 }
113
114 int main()
115 {
116     ofstream fout;
117     string line;
118
119     string FileName1 = "C:/Users/sabob/projects/ai_programming_playground_2024/ai_12/oleksandr_bobrov/tskyi/opic_5/F1.txt";
120     string FileName2 = "C:/Users/sabob/projects/ai_programming_playground_2024/ai_12/oleksandr_bobrov/tskyi/opic_5/F2.txt";
121
122     fout.open(FileName1);
123
124     cout << "Enter the text using 'enter' between each string, or type in -1 to exit" << endl;
125
126     while(fout)
127     {
128         getline(cin, line);
129         if(line == "-1")
130         {
131             break;
132         }
133         fout << line << endl;
134     }
135
136     CopyFileInFo(FileName1, FileName2);
137
138     cout << "Text from the first file: " << endl;
139     OutputOfAFile(FileName1);
140
141     cout << "Text from the second file: " << endl;
142     OutputOfAFile(FileName2);
143
144     fout.close();
145
146     return 0;
147 }

```

Код до завдання №3

Завдання №4

Посилання на файл програми у пул-запиті GitHub

```

1 #include <bits/stdc++.h>
2
3 using namespace std;
4
5 vector<string> StringsPliter(string &line)
6 {
7     stringstream s(line);
8     string word;
9     vector<string> SeparatedString;
10
11     while (s >> word) {
12         SeparatedString.push_back(word);
13     }
14
15     return SeparatedString;
16 }
17
18 void OutputOfAFile(string &FileName)
19 {
20     string line;
21     ifstream fin;
22
23     fin.open(FileName);
24
25     if(!fin)
26     {
27         cerr << "Error: unable to open the file";
28         return;
29     }
30
31     while(fin)
32     {
33         getline(fin, line);
34         cout << line << endl;
35     }
36
37     fin.close();
38 }
39
40 void CopyFileInFo(string &FileName1, string &FileName2)
41 {
42     string line;
43     ifstream fin;
44     ofstream fout;
45     fin.open(FileName1);
46     fout.open(FileName2);
47
48     vector<string> SeparatedString;
49     string ShortestWord;
50
51     if (!fin.is_open() || !fout.is_open()) {
52         cerr << "Error opening files!" << endl;
53         return;
54     }
55
56     while(getline(fin, line))
57     {
58         SeparatedString = StringsPliter(line);
59
60         if(!SeparatedString.empty())
61         {
62             ShortestWord = SeparatedString[0];
63
64             for(int i = 0; i < SeparatedString.size(); i++)
65             {
66                 if(SeparatedString[i].length() < ShortestWord.length())
67                 {
68                     ShortestWord = SeparatedString[i];
69                 }
70             }
71         }
72     }
73
74     int count = 0;
75     int SkippedLine = -1;
76
77     fin.clear();
78     fin.seek(0, ios::beg);
79
80     while(fin)
81     {
82         getline(fin, line);
83         SeparatedString = StringsPliter(line);
84         count++;
85
86         for(int i = 0; i < SeparatedString.size(); i++)
87         {
88             if(SeparatedString[i] == ShortestWord)
89             {
90                 SkippedLine = count;
91             }
92         }
93     }
94
95     count = 0;
96
97     fin.clear();
98     fin.seek(0, ios::beg);
99
100     while(getline(fin, line))
101     {
102         count++;
103         if(count != SkippedLine)
104         {
105             fout << line << endl;
106         }
107     }
108
109     fin.close();
110     fout.close();
111 }
112
113 int main()
114 {
115     ofstream fout;
116     string line;
117
118     string FileName1 = "C:/Users/sabob/projects/ai_programming_playground_2024/ai_12/oleksandr_bobrov/tskyi/opic_5/F1.txt";
119     string FileName2 = "C:/Users/sabob/projects/ai_programming_playground_2024/ai_12/oleksandr_bobrov/tskyi/opic_5/F2.txt";
120
121     fout.open(FileName1);
122
123     cout << "Enter the text using 'enter' between each string, or type in -1 to exit" << endl;
124
125     while(fout)
126     {
127         getline(cin, line);
128         if(line == "-1")
129         {
130             break;
131         }
132         fout << line << endl;
133     }
134
135     CopyFileInFo(FileName1, FileName2);
136
137     cout << "Text from the first file: " << endl;
138     OutputOfAFile(FileName1);
139
140     cout << "Text from the second file: " << endl;
141     OutputOfAFile(FileName2);
142
143     fout.close();
144
145     return 0;
146 }

```

Код до завдання №4

Завдання №5

Посилання на файл програми у пул-запиті GitHub


```

1 #include<bits/stdc++.h>
2
3 using namespace std;
4
5 vector<int> Difference(vector<int> &v1, vector<int> &v2)
6 {
7     vector<int> result;
8
9     for(int i = 0; i < v1.size(); i++)
10     {
11         if(find(v2.begin(), v2.end(), v1[i]) == v2.end())
12         {
13             result.push_back(v1[i]);
14         }
15     }
16
17     sort(result.begin(), result.end());
18
19     return result;
20 }
21
22 vector<int> ArrayIntersection(vector<int> v1, vector<int> v2)
23 {
24     vector<int> result;
25
26     for(int i = 0; i < v1.size(); i++)
27     {
28         auto FindPosition = find(v2.begin(), v2.end(), v1[i]);
29         if(FindPosition != v2.end())
30         {
31             result.push_back(v1[i]);
32             v1[i] = -1;
33             *FindPosition = -1;
34         }
35     }
36
37     sort(result.begin(), result.end());
38
39     return result;
40 }
41
42 vector<int> ArrayUnion(vector<int> v1, vector<int> v2)
43 {
44     vector<int> result = v1;
45
46     for(int i = 0; i < v2.size(); i++)
47     {
48         if(find(v1.begin(), v1.end(), v2[i]) == v1.end())
49         {
50             result.push_back(v2[i]);
51         }
52     }
53
54     sort(result.begin(), result.end());
55
56     return result;
57 }
58
59 vector<int> SymmetricDifference(vector<int> v1, vector<int> v2)
60 {
61     vector<int> result;
62     for (int value : v1)
63     {
64         if (find(v2.begin(), v2.end(), value) == v2.end())
65         {
66             result.push_back(value);
67         }
68     }
69
70     for (int value : v2)
71     {
72         if (find(v1.begin(), v1.end(), value) == v1.end())
73         {
74             result.push_back(value);
75         }
76     }
77
78     sort(result.begin(), result.end());
79
80     return result;
81 }
82
83 int main()
84 {
85     int n, m;
86     cin >> n;
87     if(n < 1 || n > 100)
88     {
89         return 1;
90     }
91
92     vector<int> num1(n);
93
94     for(int i = 0; i < n; i++)
95     {
96         cin >> num1[i];
97         if(num1[i] < 1 || num1[i] > 100)
98         {
99             return 1;
100         }
101     }
102
103     sort(num1.begin(), num1.end());
104
105     cin >> m;
106     if(m < 1 || m > 100)
107     {
108         return 1;
109     }
110     vector<int> num2(m);
111
112     for(int i = 0; i < m; i++)
113     {
114         cin >> num2[i];
115         if(num2[i] < 1 || num2[i] > 100)
116         {
117             return 1;
118         }
119     }
120
121     sort(num2.begin(), num2.end());
122
123     vector<int> R_H_Difference = Difference(num1, num2);
124     vector<int> R_H_Difference = Difference(num2, num1);
125     vector<int> Intersection = ArrayIntersection(num1, num2);
126     vector<int> Union = ArrayUnion(num1, num2);
127     vector<int> XOR = SymmetricDifference(num1, num2);
128
129     cout << R_H_Difference.size() << endl;
130     for(int i : R_H_Difference)
131     {
132         cout << i << " ";
133     }
134     cout << "\n\n";
135
136     cout << R_H_Difference.size() << endl;
137     for(int i : R_H_Difference)
138     {
139         cout << i << " ";
140     }
141     cout << "\n\n";
142
143     cout << Intersection.size() << endl;
144     for(int i : Intersection)
145     {
146         cout << i << " ";
147     }
148     cout << "\n\n";
149
150     cout << Union.size() << endl;
151     for(int i : Union)
152     {
153         cout << i << " ";
154     }
155     cout << "\n\n";
156
157     cout << XOR.size() << endl;
158     for(int i : XOR)
159     {
160         cout << i << " ";
161     }
162
163     return 0;
164 }
165
166

```

Код до завдання №5

Завдання №6

Посилання на файл програми у пул-запиті GitHub

```

1  #include<bits/stdc++.h>
2
3  using namespace std;
4
5  vector<int> PossibleValues(vector<vector<int>> &v1, int x, int y, int n)
6  {
7      vector<int> result;
8      if(v1[x][y] != 0)
9      {
10         result.push_back(v1[x][y]);
11         return result;
12     }
13
14     set<int> ForbiddenValues;
15
16     for(int i = 0; i < n; i++)
17     {
18         if(v1[i][y] != 0)
19         {
20             ForbiddenValues.insert(v1[i][y]);
21         }
22     }
23
24     for(int i = 0; i < n; i++)
25     {
26         if(v1[x][i] != 0)
27         {
28             ForbiddenValues.insert(v1[x][i]);
29         }
30     }
31
32     for(int i = 1; i <= n; i++)
33     {
34         if(ForbiddenValues.find(i) == ForbiddenValues.end())
35         {
36             result.push_back(i);
37         }
38     }
39
40     return result;
41 }
42
43 int main()
44 {
45     int N, Q, x, y = 0;
46     string row;
47     cin >> N;
48     if(N > 9 || N < 1)
49     {
50         return 0;
51     }
52
53     vector<vector<int>> v1(N, vector<int>(N));
54     for(int i = 0; i < N; i++)
55     {
56         cin >> row;
57         if(row.length() != N)
58         {
59             return 0;
60         }
61         for(int j = 0; j < N; j++)
62         {
63             v1[i][j] = row[j] - 48;
64             if(v1[i][j] > 9 || v1[i][j] < 0)
65             {
66                 return 0;
67             }
68         }
69     }
70
71     cin >> Q;
72
73     if(Q > 1000 || Q < 1)
74     {
75         return 0;
76     }
77
78     vector<pair<int, int>> position;
79     for(int i = 0; i < Q; i++)
80     {
81         cin >> x >> y;
82         x--;
83         y--;
84         position.push_back({x, y});
85     }
86
87     for(int i = 0; i < Q; i++)
88     {
89         vector<int> possible = PossibleValues(v1, position[i].first, position[i].second, N);
90         int s = possible.size();
91         cout << s << endl;
92         for(int j = 0; j < s; j++)
93         {
94             cout << possible[j] << " ";
95         }
96         cout << endl;
97     }
98
99     return 0;
100 }
101

```

Код до завдання №6

Завдання №7

Посилання на файл програми у пул-запиті GitHub

```

1 #include<bits/stdc++.h>
2
3 using namespace std;
4
5 enum FileOpResult{Success, Failure};
6
7 FileOpResult write_to_file(const char 'name, char 'content)
8 {
9     ifstream fin;
10
11     string line;
12
13     string StringContent(content);
14
15     fin.open(name);
16
17     if(!fin.is_open())
18     {
19         cerr << "Error: unable to open the file";
20         return Failure;
21     }
22
23     if(getline(fin, line))
24     {
25         if(line != StringContent)
26         {
27             cerr << "Error: input is not equal to file content";
28             return Failure;
29         }
30     }
31     else
32     {
33         cerr << "Error: file is clear";
34         return Failure;
35     }
36
37     fin.close();
38
39     return Success;
40 }
41
42 FileOpResult copy_file(const char 'file_from, const char 'file_to)
43 {
44     ofstream fto;
45     ifstream ffrom;
46
47     vector<string> lines_from;
48     vector<string> lines_to;
49
50     string line;
51
52     fto.open(file_to);
53     ffrom.open(file_from);
54     if(!ffrom.is_open())
55     {
56         cerr << "Error: unable to open the file 'to' << endl;
57         return Failure;
58     }
59
60     if(!ffrom.is_open())
61     {
62         cerr << "Error: unable to open the file 'from' << endl;
63         return Failure;
64     }
65
66     while(getline(ffrom, line))
67     {
68         fto << line;
69     }
70
71     fto.close();
72     ffrom.close();
73
74     ifstream ftoCheck;
75     ifstream ffromCheck;
76
77     ftoCheck.open(file_to);
78     ffromCheck.open(file_from);
79
80     while(getline(ffromCheck, line))
81     {
82         lines_from.push_back(line);
83     }
84
85     while(getline(ftoCheck, line))
86     {
87         lines_to.push_back(line);
88     }
89
90     if(lines_from.size() == 0)
91     {
92         cerr << "Error: file 'from' is clear" << endl;
93         return Failure;
94     }
95
96     if(lines_to.size() == 0)
97     {
98         cerr << "Error: file 'to' is clear" << endl;
99         return Failure;
100     }
101
102     for(int i = 0; i < lines_from.size(); i++)
103     {
104         if(lines_from[i] != lines_to[i])
105         {
106             cerr << "Error: contents of the two files are not equal" << endl;
107             return Failure;
108         }
109     }
110
111     ftoCheck.close();
112     ffromCheck.close();
113
114     return Success;
115 }
116
117 int main()
118 {
119     ofstream fout;
120
121     const char 'fileName1 = "C:/Users/sabob/projects/ai_programming_playground_2024/ai_12/oleksandr_bobrovyskiy/epic_5/file_from.txt";
122     const char 'fileName2 = "C:/Users/sabob/projects/ai_programming_playground_2024/ai_12/oleksandr_bobrovyskiy/epic_5/file_to.txt";
123
124     char content[256];
125
126     fout.open(fileName1);
127
128     if(!fout.is_open())
129     {
130         cout << Failure << endl;
131         cerr << "Error: unable to open the file";
132         return 1;
133     }
134
135     cin.getline(content, 256);
136
137     fout << content;
138
139     fout.close();
140
141     cout << "Execution status of the task 1 : " << write_to_file(fileName1, content) << endl;
142     cout << "Execution status of the task 2 : " << copy_file(fileName1, fileName2);
143
144     return 0;
145 }

```

Код до завдання №7

Завдання №8

Посилання на файл програми у пул-запиті GitHub

```
1  #include<bits/stdc++.h>
2
3  using namespace std;
4
5  int main()
6  {
7      int left, count = 1;
8      int right = 1e9;
9      int middle;
10     string response;
11
12     while (left <= right && count !=48)
13     {
14         middle = left + (right - left) / 2;
15         cout << middle << endl << flush;
16
17         cin >> response;
18
19         if(response != "=" && response != ">" && response != "<")
20         {
21             continue;
22         }
23
24         if (response == "=")
25         {
26             break;
27         }
28         else if (response == ">")
29         {
30             right = middle - 1;
31         }
32         else if (response == "<")
33         {
34             left = middle + 1;
35         }
36         count++;
37     }
38
39     return 0;
40 }
41
```

Код до завдання №8

4. Результати виконання завдань, тестування та фактично затрачений час:

Завдання №1 Деталі по виконанню і тестуванню програми

```
print a line with a number that should be an integer between  
tnirp a enil htiw a rebmun taht dluohs eb na regetni neewteb
```

Блок №1 Результат виконання завдання

Час затрачений на виконання завдання – 20min

Завдання №2 Деталі по виконанню і тестуванню програми

```
Enter the max price:
```

```
15
```

```
Film1
```

```
Director1
```

```
150
```

```
15
```

```
Film2
```

```
Director2
```

```
125
```

```
20
```

```
Film3
```

```
Director3
```

```
90
```

```
10
```

```
Film4
```

```
Director4
```

```
100
```

```
25
```

```
Add 3 more videotapes
Enter the new name of the videotape
Film5
Enter the new name of the director
Director5
Enter the new running time of the film
99
Enter the new price of the videotape
19
Enter the new name of the videotape
Film6
Enter the new name of the director
director6
Enter the new running time of the film
130
Enter the new price of the videotape
12
Enter the new name of the videotape
film7
Enter the new name of the director
director7
Enter the new running time of the film
50
Enter the new price of the videotape
5
```

New List of videotapes:

```
Film1
Director1
150
15
```

```
Film3
Director3
90
10
```

```
Film5
Director5
99
19
```

```
Film6
director6
130
12
```

```
film7
director7
50
5
```

Блок №1 Результат виконання завдання

Час затрачений на виконання завдання – 35min

Завдання №3 Деталі по виконанню і тестуванню програми


```

Enter the text using 'Enter' between each string, or type in -1 to exit
Blue
Running fast
Silent night
Bright
Happy thoughts
Dancing shadows
Golden dreams
Crisp
Gentle breeze
Flowing river calm
Endless horizon expanding wide
-1
Text from the first file:
Blue
Running fast
Silent night
Bright
Happy thoughts
Dancing shadows
Golden dreams
Crisp
Gentle breeze
Flowing river calm
Endless horizon expanding wide
Text from the second file:
Blue
Running fast
Silent night
Bright
Happy thoughts
Dancing shadows
Golden dreams
Crisp
Gentle breeze
Flowing river calm

```

Блок №1 Результат виконання завдання

Час затрачений на виконання завдання – 45min

Завдання №4 Деталі по виконанню і тестуванню програми

```

5
1 2 3 4 5
5
4 5 6 7 8

3
1 2 3

3
6 7 8

2
4 5

8
1 2 3 4 5 6 7 8

6
1 2 3 6 7 8

```

Accepted
Wrong Answer 2
Wrong Answer 2
Wrong Answer 2
Wrong Answer 2
Wrong Answer 2
Wrong Answer 2

Блок №1 Результат виконання завдання

Час затрачений на виконання завдання – 20min

Завдання №5 Деталі по виконанню і тестуванню програми

```

5
1 2 3 4 5
5
4 5 6 7 8
3
1 2 3

3
6 7 8

2
4 5

8
1 2 3 4 5 6 7 8

6
1 2 3 6 7 8

```

Блок №1 Результат виконання завдання

Час затрачений на виконання завдання – 40min

Завдання №6 Деталі по виконанню і тестуванню програми

```

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

```

Блок №1 Результат виконання завдання

Час затрачений на виконання завдання – 30min

Завдання №7 Деталі по виконанню і тестуванню програми

```
trying to find a way to test whether my class has properly closed
Execution status of the task 1 : 0
Execution status of the task 2 : 0
```

Блок №1 Результат виконання завдання

Час затрачений на виконання завдання – 35min

Завдання №8 Деталі по виконанню і тестуванню програми

```
5000000000
>
2499999999
>
1249999999
>
624999999
>
312499999
>
156249999
>
7812499
>
```

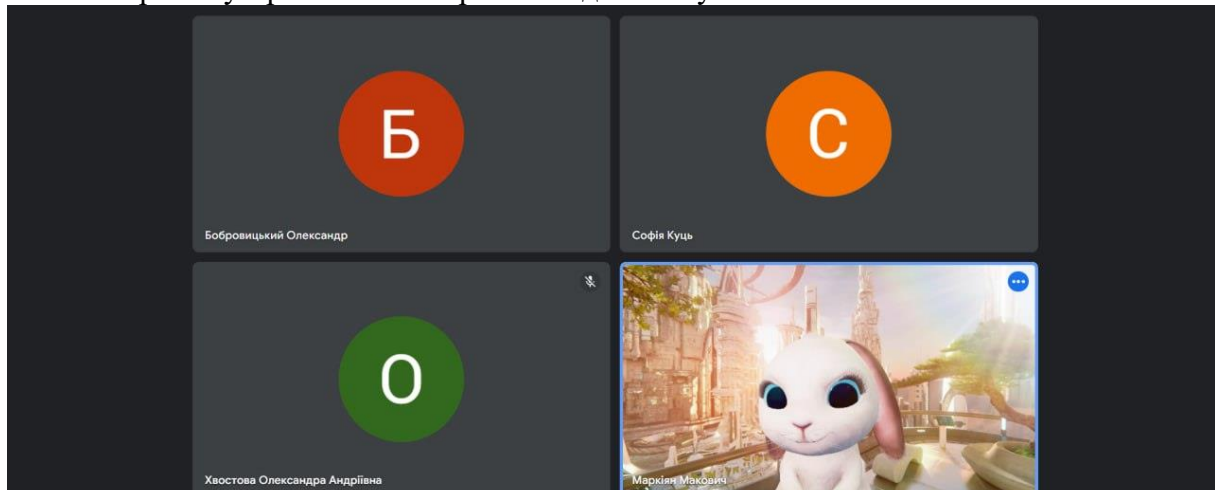
```
28
<
43
<
50
>
46
<
48
>
47
=
```

Блок №1 Результат виконання завдання

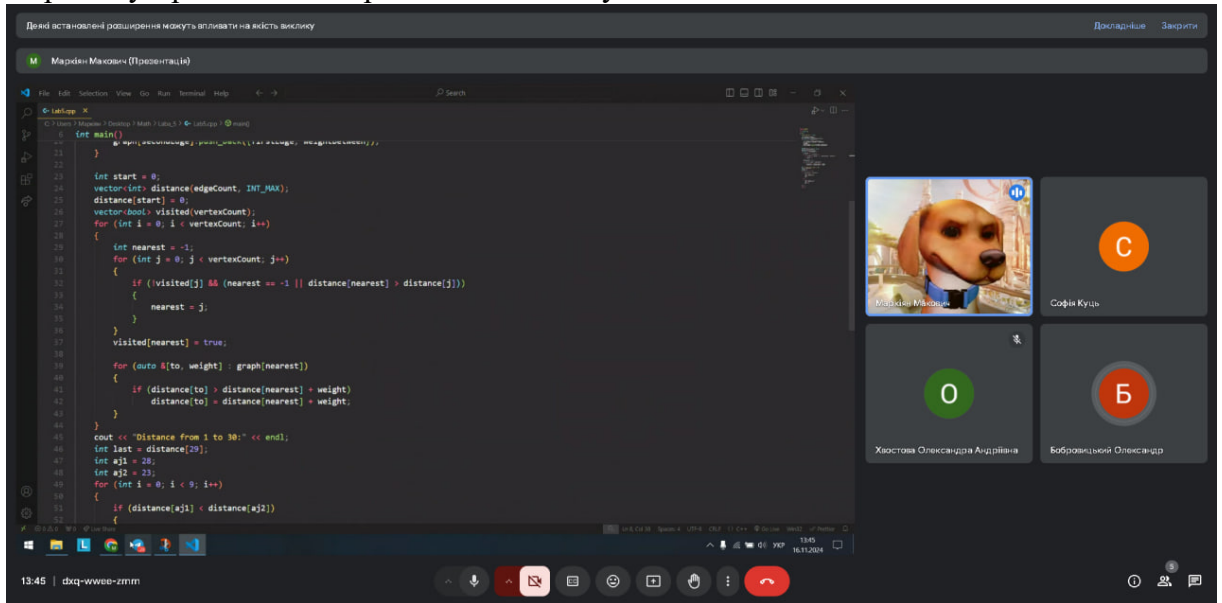
Час затрачений на виконання завдання – 20min

5. Кооперація з командою:

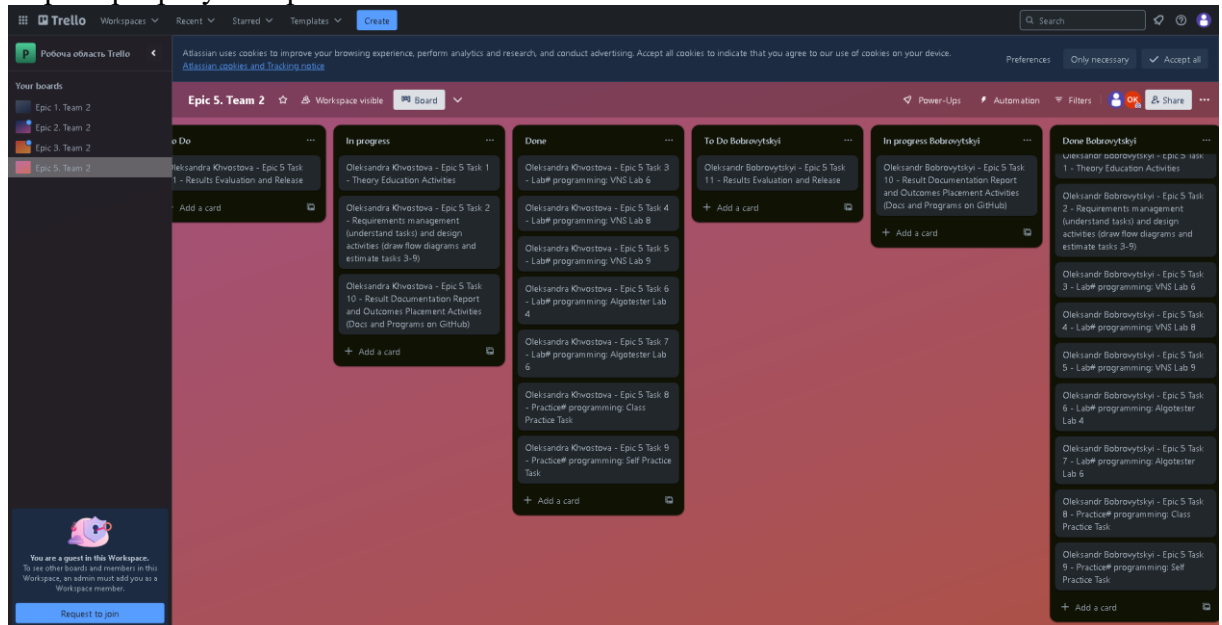
– Скрін з зустрічі по обговоренню задач Епіку 1



– Скрін з зустрічі по обговоренню задач Епіку 2



– Скрін прогресу по Трелю



Висновок: я ознайомився з темами лабораторної, опрацювати їх теоретично та навчитися використовувати отриманні знання для вирішення практичних задач.