

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

Виконала:

Студентка групи ІІІ-12
Хвостова Олександра Андріївна

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:

Ознайомитися з різними аспектами роботи з файлами у C++ та здобути практичні навички у виконанні основних операцій з файлами, роботи з файловими дескрипторами та C-style читання і запису. Також метою було розібратися з роботою з char та string, стрічковими літералами, екрануванням символів, конкатенацією, порівнянням та пошуком у рядках. Важливо було зрозуміти особливості читання та запису текстових файлів, парсинг текстових файлів, форматування тексту при записі, обробку помилок, а також відмінності між бінарними та текстовими файлами. Також передбачалося ознайомитися з оглядом стандартної бібліотеки для роботи з файлами та потоками вводу/виводу. Нарешті, метою було ознайомитися зі створенням власних бібліотек у C++, їх документуванням та використанням сторонніх бібліотек у проєктах.

Теоретичні відомості:

1. Вступ до Роботи з Файлами:

- Джерела Інформації
 - Стаття. <https://acode.com.ua/urok-220-bazovyy-fajlovyj-vvid-i-vyvid/>
 - Стаття. <https://www.youtube.com/watch?v=FeNqHytI0fA&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=166>
- Що опрацьовано:
 - Основні операції з файлами: відкриття, читання, запис, закриття
 - Робота з файловими дескрипторами
 - C-style читання з файлу та запис до файлу
 - Перевірка стану файлу: перевірка помилок, кінець файлу
 - Базові приклади читання та запису в файл
- Статус: Ознайомлений
- Початок опрацювання теми: 12.11.2024
- Завершення опрацювання теми: 12.11.2024

2. Символи і Рядкові Змінні:

- Джерела Інформації
 - Стаття. <https://acode.com.ua/urok-38-symvolnyj-typ-danyh-char/>
 - Стаття. <https://acode.com.ua/urok-39-litaly-i-magichni-chysla/>
 - Стаття. https://w3schoolsua.github.io/cpp/cpp_strings_concat.html#gsc.tab=0
 - Стаття. <https://acode.com.ua/urok-208-ryadkovi-klasy-std-string-i-std-wstring/>
 - Стаття. <http://www.kytok.org.ua/post/cplusplus-std-search>
- Що опрацьовано:
 - Робота з char та string: основні операції і методи
 - Стрічкові літерали та екранування символів
 - Конкатенація, порівняння та пошук у рядках
- Статус: Ознайомлений
- Початок опрацювання теми: 13.11.2024
- Завершення опрацювання теми: 13.11.2024

3. Текстові Файли:

- Джерела Інформації
 - Стаття. <https://itproger.com/ua/spravka/cpp/getline>
 - Стаття. <https://acode.com.ua/urok-216-funktsional-klasu-istream/>
 - Стаття. <https://en.cppreference.com/w/cpp/io/manip/setfill>
 - Стаття. <https://cplusplus.com/reference/iostream/setprecision/>
- Що опрацьовано:
 - Особливості читання та запису текстових файлів
 - Обробка рядків з файлу: getline, ignore, peek
 - Форматування тексту при записі: setw, setfill, setprecision
 - Парсинг текстових файлів: розділення на слова, аналіз структури
 - Обробка помилок при роботі з файлами
- Статус: Ознайомлений
- Початок опрацювання теми: 14.11.2024
- Завершення опрацювання теми: 14.11.2024

4. Бінарні Файли:

- Джерела Інформації
 - Стаття. <https://abitap.com/6-0-tekstovi-ta-binarni-fajly/>
 -
- Що опрацьовано:
 - Вступ до бінарних файлів: відмінності від текстових, приклади (великі дані, ігрові ресурси, зображення)
 - Читання та запис бінарних даних
 - Робота з позиціонуванням у файлі: seekg, seekp
 - Сериалізація об'єктів у бінарний формат
- Статус: Ознайомлений
- Початок опрацювання теми: 14.11.2024
- Завершення опрацювання теми: 14.11.2024

5. Стандартна бібліотека та робота з файлами:

- Джерела Інформації
 - Стаття. <https://acode.com.ua/urok-220-bazoviy-fajlovyj-vvid-i-vyvvid/>
 -
- Що опрацьовано:
 - Огляд стандартної бібліотеки для роботи з файлами
 - Потоки вводу/виводу: ifstream, ofstream, fstream
 - Обробка помилок при роботі з файлами
- Статус: Ознайомлений
- Початок опрацювання теми: 14.11.2024
- Завершення опрацювання теми: 14.11.2024

6. Створення й використання бібліотек:

- Джерела Інформації
 - Відео. <https://www.youtube.com/watch?v=mnwDpO4zqLA>
- Що опрацьовано:
 - Вступ до створення власних бібліотек у C++
 - Правила розбиття коду на header-и(.h) та source(.cpp) файли
 - Статичні проти динамічних бібліотек: переваги та використання
 - Інтерфейси бібліотек: створення, документування, версіонування
 - Використання сторонніх бібліотек у проектах
- Статус: Ознайомлений
- Початок опрацювання теми: 14.11.2024

- Завершення опрацювання теми: 14.11.2024

Виконання роботи:

1. Опрацювання завдання та вимог до програм та середовища:

Завдання №1. VNS Lab 6 - Task 1

- Варіант завдання: 7
- Планований час на виконання: 1 година
- Деталі завдання

Перетворити рядок так, щоб всі цифри в ньому були відсортовані за спаданням.

- Важливі деталі для врахування в імплементації програми

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.

Завдання №2. VNS Lab 8 - Task 1

- Варіант завдання: 7
- Планований час на виконання: 3 години
- 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.

- Важливі деталі для врахування в імплементації програми

Структура "Студент":

- прізвище, ім'я, по батькові;
- домашня адреса;
- група;
- рейтинг.

Знищити всі елементи, у яких рейтинг менше заданого, додати 1 елемент у кінець файлу.

Завдання №3. VNS Lab 9 - Task 1

- Варіант завдання: 7
- Планований час на виконання: 4 години
- Деталі завдання: Створити текстовий файл F1 не менше, ніж з 10 рядків і записати в нього інформацію
- Важливі деталі для врахування в імплементації програми:
 - 1) Скопіювати з файлу F1 у файл F2 всі рядки, крім тих, що починаються на букву А.
 - 2) Підрахувати кількість символів у першому слові F2.

Завдання №4-5. Algotester - Лабораторна робота №4

- Варіант завдання: 3
- Планований час на виконання: 2 години
- Деталі завдання:

Обмеження: 2 сек., 256 MiB

Вам дано масив, який складається з N додатніх 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 нарешті 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 видаліть усі дублікати з масиву.

Виведіть результуючий масив.

- Важливі деталі для врахування в імплементації програми:

Вхідні дані

У першому рядку N - кількість чисел.

У другому рядку N чисел a_i - елементи масиву.

Вихідні дані

У першому рядку M - кількість чисел у масиву

У другому рядку M посортованих за умовою чисел.

Обмеження

$$1 \leq N \leq 10^3$$

$$0 \leq a_i \leq 10^3$$

Завдання №6. Algotester - Лабораторна робота №6 – 3 варіант

- Варіант завдання: 3
- Планований час на виконання: 4 години
- 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 пусті (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.

- Важливі деталі для врахування в імплементації програми:

Вхідні дані

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 запиту

Вихідні дані

Q разів відповідь у наступному форматі:

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

M цифр розділених пробілом - можливі цифри

Завдання №7. Self Practice

- Планований час на виконання: 1 година
- Деталі завдання:

Сортування

Обмеження: 2 сек., 256 МБ

Масив a складається з n цілих невід'ємних чисел. Потрібно його посортувати.

Вхідні дані

У єдиному рядку задано елементи масиву a через пробіл.

Вихідні дані

У єдиному рядку виведіть посортований масив. Елементи масиву розділіть пробілами.

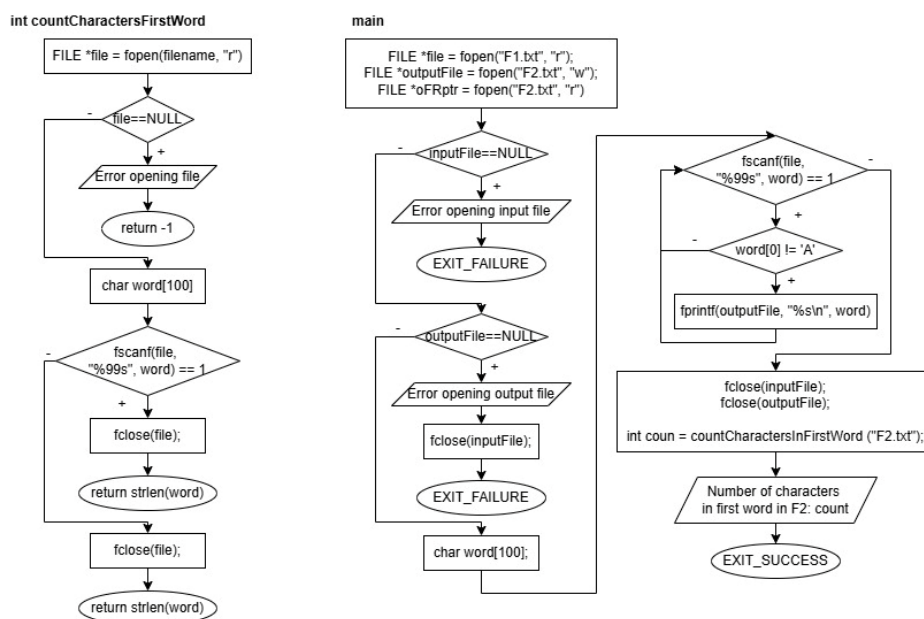
Обмеження

$$1 \leq n \leq 10^4,$$

$$0 \leq a_i \leq 5 \cdot 10^4.$$

2. Дизайн та планована оцінка часу виконання завдань:

Завдання №3. VNS Lab 9 - Task 1



4. Код програм з посиланням на зовнішні ресурси:

Завдання №__ Деталі по програмі + Вставка з кодом з підписами до вставки. Посилання на файл програми у пул-запиті GitHub

Завдання №1. VNS Lab 6 - Task 1

- Варіант завдання: 7

```
1  #include <iostream>
2  #include <cstring> // Для strtok
3  #include <algorithm> // Для sort
4
5  using namespace std;
6
7  int main() {
8      char str[255];
9      cout << "Enter a string of numbers separated by spaces: ";
10     gets(str);
11
12     char* token = strtok(str, " ");
13     int numbers[255];
14     int count = 0;
15
16     while (token != NULL) {
17         numbers[count++] = atoi(token);
18         token = strtok(NULL, " ");
19     }
20
21     sort(numbers, numbers+count);
22
23     cout << "Sorted numbers: ";
24     for (int i = count-1; i >0; i--) {
25         cout << numbers[i] << " ";
26     }
27     cout << endl;
28     delete[] token;
29
30     return 0;
31 }
```

Завдання №2. VNS Lab 8 - Task 1

- Варіант завдання: 7

```
1  #include <iostream>
2  #include <cstdio>
3  #include <cstdlib>
4  #include <cstring>
5
6  using namespace std;
7
8  typedef struct {
9      char name[50];
10     char address[50];
11     char group[50];
12     int rate;
13 } Student;
14
15 void addInitialData(const char* filename);
16 void Delete(const char* filename, int rate);
17 void printFile(const char* filename);
18 void addNew(const char* filename, const Student& newStudent); // Fixed function declaration
19
20 int main() {
21     const char* filename = "students.dat";
22     addInitialData(filename);
23     printFile(filename);
24
25     int rate;
26     cout << "Enter the minimum rate to keep students: ";
27     cin >> rate;
28
29     Delete(filename, rate);
30     printFile(filename);
31
32     Student newStudent = {"NewName", "NewAddress", "NewGroup", 88};
33     addNew(filename, newStudent);
34     printFile(filename);
35
36     return 0;
37 }
38
39 void addInitialData(const char* filename) {
40     FILE* f = fopen(filename, "wb");
41     if (f == NULL) {
42         cerr << "Can't open the file." << endl;
43         exit(1);
44     }
45
46     Student listOfStudents[] = {
47         {"Name1", "Address1", "Group1", 21},
48         {"Name2", "Address2", "Group2", 56},
49         {"Name3", "Address3", "Group3", 43},
50         {"Name4", "Address4", "Group4", 20},
51         {"Name5", "Address5", "Group5", 28},
52         {"Name6", "Address6", "Group6", 93}
53     };
54
55     for (int i = 0; i < 6; i++) {
56         if (fwrite(&listOfStudents[i], sizeof(Student), 1, f) != 1) {
57             cerr << "Error writing to file." << endl;
58             exit(2);
59         }
60     }
61     fclose(f);
62     cout << "All students are added successfully." << endl;
63 }
```



```

65 void Delete(const char* filename, int rate) {
66     FILE* fPtr = fopen(filename, "rb");
67     if (fPtr == NULL) {
68         cout << "Error opening file." << endl;
69         return;
70     }
71
72     Student* fEms = new Student[6];
73     fread(fEms, sizeof(Student), 6, fPtr);
74     fclose(fPtr);
75
76     // Count students with rating >= rate
77     size_t cNewEms = 0;
78     for (size_t i = 0; i < 6; i++) {
79         if (fEms[i].rate >= rate) {
80             cNewEms++;
81         }
82     }
83
84     Student* dEms = new Student[cNewEms];
85     size_t dEmsI = 0;
86     for (size_t i = 0; i < 6; i++) {
87         if (fEms[i].rate >= rate) {
88             dEms[dEmsI++] = fEms[i];
89         }
90     }
91
92     FILE* fnPtr = fopen(filename, "wb");
93     if (fnPtr == NULL) {
94         cout << "Error opening file." << endl;
95         delete[] fEms;
96         delete[] dEms;
97         return;
98     }
99     fwrite(dEms, sizeof(Student), cNewEms, fnPtr);
100    fclose(fnPtr);

```

```

101
102    delete[] fEms;
103    delete[] dEms;
104 }
105
106 void printFile(const char* filename) {
107     FILE* f = fopen(filename, "rb");
108     if (f == NULL) {
109         cerr << "Error opening file." << endl;
110         exit(5);
111     }
112
113     Student person;
114     cout << "Current list of students:" << endl;
115     cout << "-----" << endl;
116
117     while (fread(&person, sizeof(Student), 1, f) == 1) {
118         cout << "Name: " << person.name << endl;
119         cout << "Address: " << person.address << endl;
120         cout << "Group: " << person.group << endl;
121         cout << "Rate: " << person.rate << endl;
122         cout << "-----" << endl;
123     }
124
125     fclose(f);
126 }
127
128 void addNew(const char* filename, const Student& newStudent) {
129     // Open the file for reading
130     FILE* fPtr = fopen(filename, "rb");
131     if (fPtr == NULL) {
132         cout << "Error opening file." << endl;
133         return;
134     }

```

```

136 // Determine the number of existing records
137 fseek(fPtr, 0, SEEK_END);
138 long fileSize = ftell(fPtr);
139 size_t numRecords = fileSize / sizeof(Student);
140 fseek(fPtr, 0, SEEK_SET);
141
142 // Read the existing students
143 Student* emps = new Student[numRecords];
144 fread(emps, sizeof(Student), numRecords, fPtr);
145 fclose(fPtr);
146
147 // Create a new array with one additional slot
148 Student* newStudents = new Student[numRecords + 1];
149 for (size_t i = 0; i < numRecords; i++) {
150     newStudents[i] = emps[i];
151 }
152 newStudents[numRecords] = newStudent;
153
154 // Write the new array back to the file
155 FILE* fCptr = fopen(filename, "wb");
156 if (fCptr == NULL) {
157     cout << "Error opening file." << endl;
158     delete[] emps;
159     delete[] newStudents;
160     return;
161 }
162 fwrite(newStudents, sizeof(Student), numRecords + 1, fCptr);
163 fclose(fCptr);
164
165 delete[] emps;
166 delete[] newStudents;
167 }

```

Завдання №3. VNS Lab 9 - Task 1

- Варіант завдання: 7

```

1 #include <stdio.h>
2 #include <ctype.h>
3 #include <stdlib.h>
4 #include <string.h>
5
6 int countCharactersInFirstWord(const char* filename) {
7     FILE *file = fopen(filename, "r");
8     if (file == NULL) {
9         perror("Error opening file");
10        return -1;
11    }
12
13    char word[100];
14    if (fscanf(file, "%99s", word) == 1) {
15        fclose(file);
16        return strlen(word);
17    }
18
19    fclose(file);
20    return 0; // No words found
21 }
22
23
24 int main() {
25     FILE *inputFile = fopen("F1.txt", "r");
26     FILE *outputFile = fopen("F2.txt", "w");
27     FILE *oFRptr = fopen("F2.txt", "r");
28
29     if (inputFile == NULL) {
30         perror("Error opening input file");
31         return EXIT_FAILURE;
32     }
33     if (outputFile == NULL) {
34         perror("Error opening output file");
35         fclose(inputFile);
36         return EXIT_FAILURE;
37     }

```

```

38
39     char word[100];
40
41     while (fscanf(inputFile, "%99s", word) == 1) {
42         if (word[0] != 'A') {
43             fprintf(outputFile, "%s\n", word);
44         }
45     }
46     fclose(inputFile);
47     fclose(outputFile);
48
49     int count = countCharactersInFirstWord("F2.txt");
50
51     printf("Number of characters in first word in F2: %u", count);
52
53
54     return EXIT_SUCCESS;
55 }

```

Завдання №4-5. Algotester - Лабораторна робота №4

```

1  #include <stdio.h>
2  #include <algorithm>
3  #include <iostream>
4
5  using namespace std;
6
7  int main() {
8      int N;
9      scanf("%d", &N);
10
11      int *array = new int[N];
12
13      for (int i = 0; i < N; i++) {
14          scanf("%d", &array[i]);
15      }
16
17      int* zero = partition(array, array + N, [](int i) { return i % 3 == 0; });
18      int* one = partition(zero, array + N, [](int i) { return i % 3 == 1; });
19
20      sort(array, zero);
21      sort(zero, one, [](int a, int b) {
22          return a > b; });
23      sort(one, array + N);
24
25      int *end = unique(array, array + N);
26      int new_size = distance(array, end);
27
28      printf("%d\n", new_size);
29      for (int i = 0; i < new_size; i++) {
30          printf("%d ", array[i]);
31      }
32
33      delete[] array;
34
35      return 0;
36 }
37

```

```

1  #include <stdio.h>
2  #include <iostream>
3
4  using namespace std;
5
6  void sortDescending(int arr[], int n) {
7      for (int i = 0; i < n - 1; i++) {
8          int max_idx = i;
9          for (int j = i + 1; j < n; j++) {
10             if (arr[j] > arr[max_idx]) {
11                 max_idx = j;
12             }
13         }
14         int temp = arr[max_idx];
15         arr[max_idx] = arr[i];
16         arr[i] = temp;
17     }
18 }
19
20 void sortAscending(int arr[], int n) {
21     for (int i = 0; i < n - 1; i++) {
22         int min_idx = i;
23         for (int j = i + 1; j < n; j++) {
24             if (arr[j] < arr[min_idx]) {
25                 min_idx = j;
26             }
27         }
28         int temp = arr[min_idx];
29         arr[min_idx] = arr[i];
30         arr[i] = temp;
31     }
32 }
33
34 int removeDuplicates(int arr[], int n) {
35     if (n == 0 || n == 1) {
36         return n;
37     }
38
39     int j = 0;
40     for (int i = 0; i < n - 1; i++) {
41         if (arr[i] != arr[i + 1]) {
42             arr[j++] = arr[i];
43         }
44     }
45     arr[j++] = arr[n - 1];
46     return j;
47 }
48

```

```

49 ▾ int main() {
50     int N;
51     scanf("%d", &N);
52
53     int *array = new int[N];
54
55 ▾     for (int i = 0; i < N; i++) {
56         scanf("%d", &array[i]);
57     }
58
59     int *remainder0 = new int[N];
60     int *remainder1 = new int[N];
61     int *remainder2 = new int[N];
62     int count0 = 0, count1 = 0, count2 = 0;
63
64 ▾     for (int i = 0; i < N; i++) {
65 ▾         if (array[i] % 3 == 0) {
66             remainder0[count0++] = array[i];
67 ▾         } else if (array[i] % 3 == 1) {
68             remainder1[count1++] = array[i];
69 ▾         } else {
70             remainder2[count2++] = array[i];
71         }
72     }
73
74     sortAscending(remainder0, count0);
75     sortDescending(remainder1, count1);
76     sortAscending(remainder2, count2);
77
78     int *result = new int[N];
79     int index = 0;
80 ▾     for (int i = 0; i < count0; i++) {
81         result[index++] = remainder0[i];
82     }
83 ▾     for (int i = 0; i < count1; i++) {
84         result[index++] = remainder1[i];
85     }
86 ▾     for (int i = 0; i < count2; i++) {
87         result[index++] = remainder2[i];
88     }
89
90     int new_size = removeDuplicates(result, index);
91
92     printf("%d\n", new_size);
93 ▾     for (int i = 0; i < new_size; i++) {
94         printf("%d ", result[i]);
95     }
96     printf("\n");
97
98     delete[] array;
99     delete[] remainder0;
100    delete[] remainder1;
101    delete[] remainder2;
102    delete[] result;
103
104    return 0;
105 }

```

Завдання №6. Algotester - Лабораторна робота №6 – 3 варіант

- Варіант завдання: 3

```
1 #include <iostream>
2 #include <vector>
3 #include <string>
4 #include <fstream>
5 using namespace std;
6
7
8 int main(){
9     int N;
10    fstream file("text.bat", fstream::binary | fstream::out);
11    cin >> N;
12    vector<std::vector<int>> sudoku(N);
13    string str;
14    for(int i = 0; i < N; i++){
15        cin >> str;
16        for(const char& ch : str) sudoku[i].push_back(ch - '0');
17    }
18    int Q;
19    cin >> Q;
20    vector<vector<int>> results(Q);
21    for(int i = 0; i < Q; i++){
22        int row, col;
23        cin >> row >> col;
24        row--;
25        col--;
26        vector<int> vec(N);
27        if(sudoku[row][col] == 0){
28            for(int j = 0; j < N; j++){
29                if(j == col || sudoku[row][j] == 0) continue;
30                vec[sudoku[row][j] - 1] = sudoku[row][j];
31            }
32            for(int j = 0; j < N; j++){
33                if(j == row || sudoku[j][col] == 0) continue;
34                vec[sudoku[j][col] - 1] = sudoku[j][col];
35            }
36
37            for(int j = 0; j < N; j++){
38                if(vec[j] != j + 1) results[i].push_back(j + 1);
39            }
40        }
41        else results[i].push_back(sudoku[row][col]);
42    }
43 }
44
45 for(vector<int>& v : results){
46     cout << v.size() << endl;
47     for(int& val : v) cout << val << " ";
48     cout << endl << endl;
49 }
50 return 0;
51 }
52
```

Завдання №7. Self Practice

```
1 #include <iostream>
2 #include <vector>
3 #include <algorithm>
4 #include <sstream>
5 using namespace std;
6
7 int main() {
8     string line;
9     getline(cin, line);
10
11    stringstream ss(line);
12    vector<int> a;
13    int number;
14
15    while (ss >> number) {
16        a.push_back(number);
17    }
18
19    sort(a.begin(), a.end());
20
21    for (int i = 0; i < a.size(); i++) {
22        if (i > 0) cout << " ";
23        cout << a[i];
24    }
25    cout << endl;
26
27    return 0;
28 }
29
```

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

Завдання №1. VNS Lab 6 - Task 1

Час затрачений на виконання завдання: 2 години

```
Enter a string of numbers separated by spaces: 123 2 1 1 1 0 567
Sorted numbers: 567 123 2 1 1 1 0
```

Завдання №2. VNS Lab 8 - Task 1

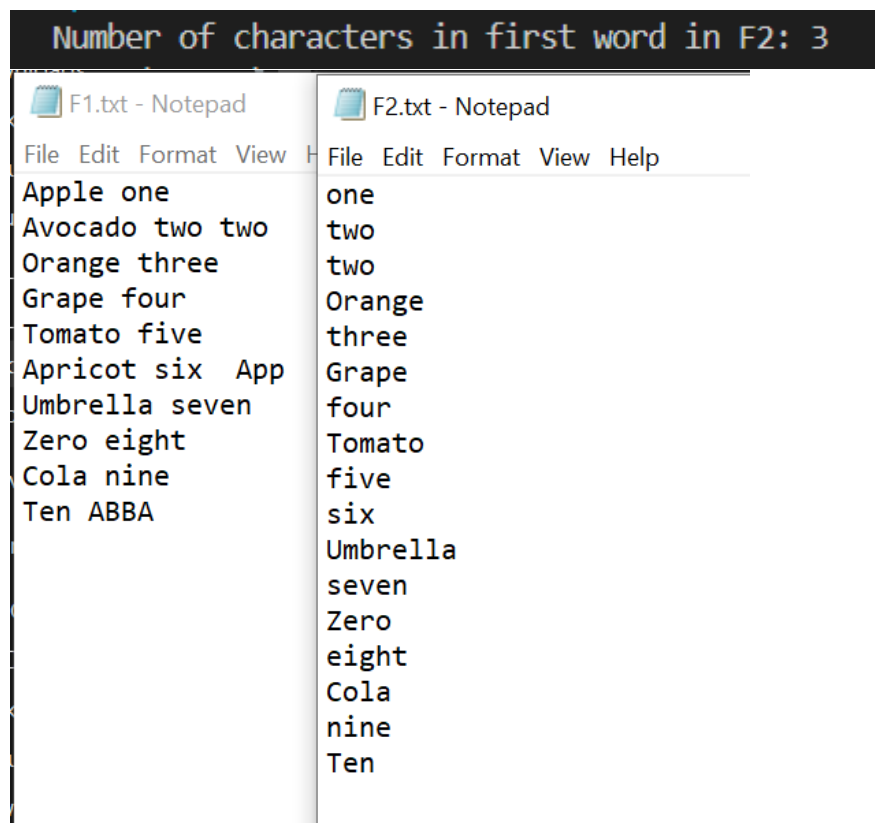
Час затрачений на виконання завдання: 5 годин

```
All students are added successfully.
Current list of students:
-----
Name: Name1
Address: Address1
Group: Group1
Rate: 21
-----
Name: Name2
Address: Address2
Group: Group2
Rate: 56
-----
Name: Name3
Address: Address3
Group: Group3
Rate: 43
-----
Name: Name4
Address: Address4
Group: Group4
Rate: 20
-----
Name: Name5
Address: Address5
Group: Group5
Rate: 28
-----
Name: Name6
Address: Address6
Group: Group6
Rate: 93
-----
```

```
Enter the minimum rate to keep students: 78
Current list of students:
-----
Name: Name6
Address: Address6
Group: Group6
Rate: 93
-----
Current list of students:
-----
Name: Name6
Address: Address6
Group: Group6
Rate: 93
-----
Name: NewName
Address: NewAddress
Group: NewGroup
Rate: 88
-----
```

Завдання №3. VNS Lab 9 - Task 1

Час затрачений на виконання завдання: 4 години



Завдання №4-5. Algotester - Лабораторна робота №4

Час затрачений на виконання завдання: 2 години

1.

```
10
1 2 3 4 4 3 2 1 1 5
5
3 4 1 2 5
```

2.

```
10
1 2 3 4 4 3 2 1 1 5
5
3 4 1 2 5
```


Завдання №6. Algotester - Лабораторна робота №6 – 3 варіант

Час затрачений на виконання завдання: 6 годин

```
2
2 3

1
2

1
1
```

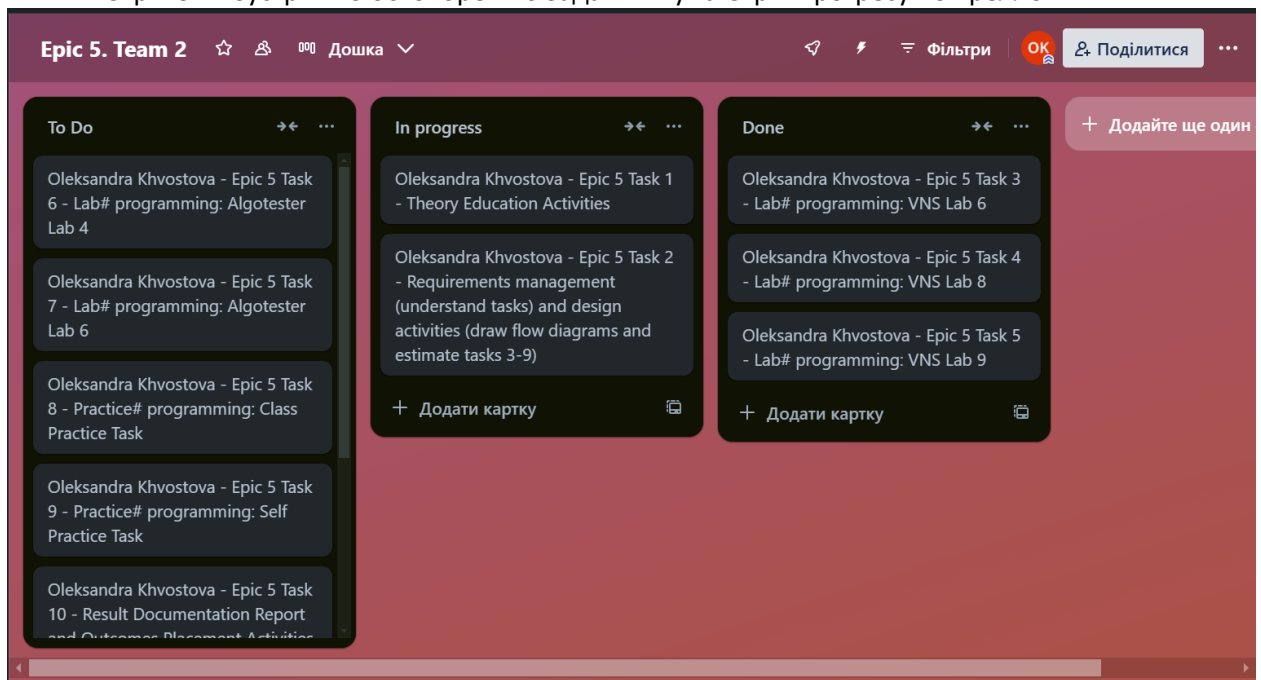
Завдання №7. Self Practice

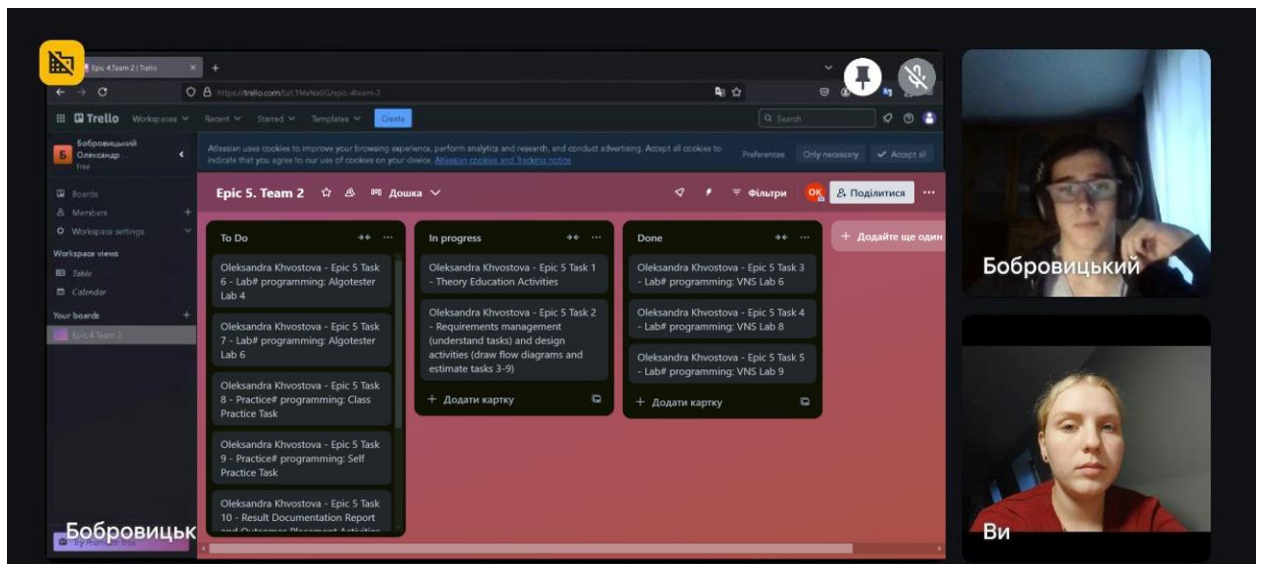
Час затрачений на виконання завдання: 2 години

```
1 3 4 2 3 6 7 8 2 1 1 3 0
0 1 1 1 2 2 3 3 3 4 6 7 8
```

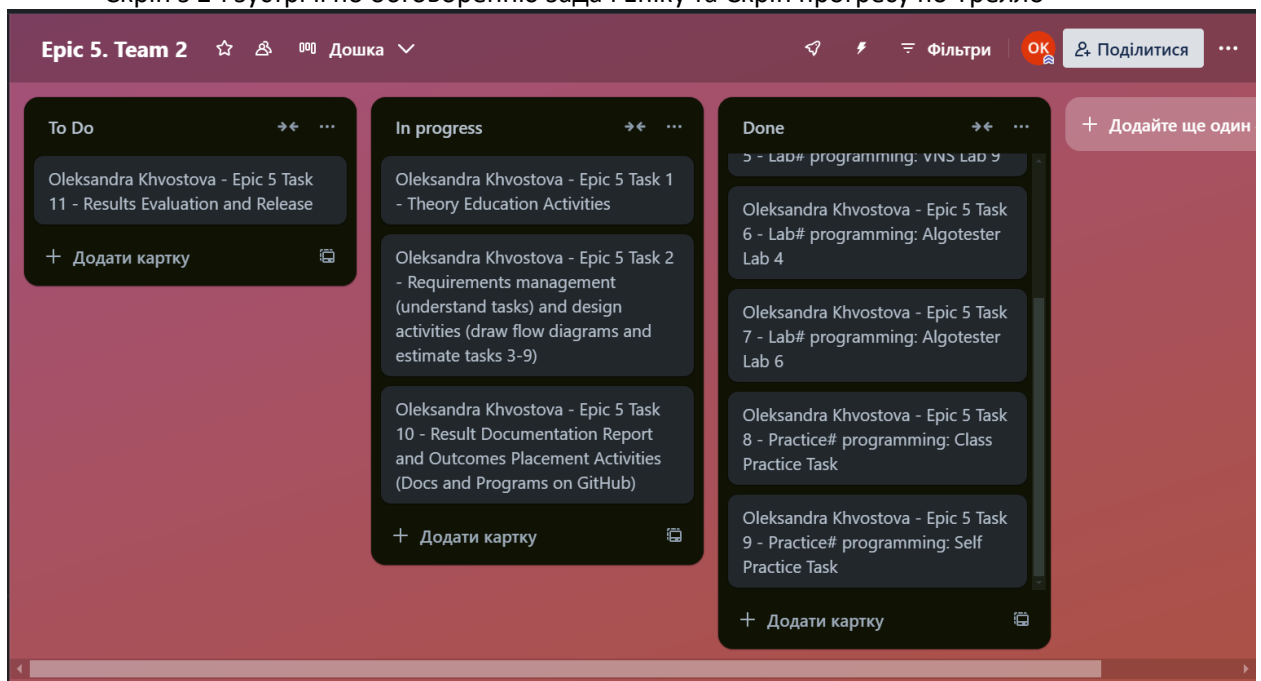
6. Кооперація з командою:

- Скрін з 1-ї зустрічі по обговоренню задач Епіку та Скрін прогресу по Тrello

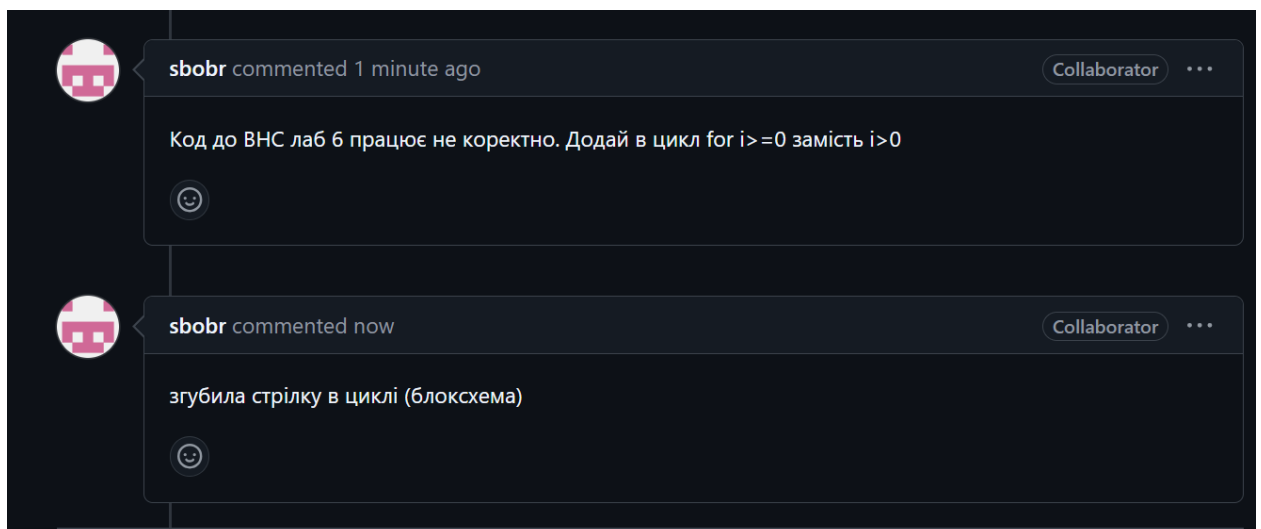




- Скрін з 2-ї зустрічі по обговоренню задач Епіку та Скрін прогресу по Трелло



Скрін з 2-му коментарями від учасників команди на пул реквести з Ревю Роботи



Висновки:

Робота з файлами є невід'ємною частиною програмування на C++, і вміння відкривати, читати, записувати та закривати файли дозволяє ефективно зберігати та обробляти дані. Знання основних операцій з файловими дескрипторами та C-style читання і запису допомагає краще розуміти низькорівневі аспекти роботи з файлами. Розуміння символів і рядкових змінних, а також методів їх обробки, є ключовим для ефективного маніпулювання текстовими даними. Текстові файли дозволяють зберігати дані у зручному для читання форматі, а методи форматування тексту дозволяють створювати структуровані файли. Бінарні файли дозволяють зберігати великі обсяги даних у компактному форматі, що особливо корисно для складних об'єктів та мультимедійних файлів. Використання стандартної бібліотеки C++ спрощує роботу з файлами завдяки зручним класам та методам для вводу/виводу. Створення власних бібліотек дозволяє розширювати функціональність програм та використовувати повторюваний код у різних проектах, що підвищує ефективність розробки.