



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 ШІ-11

Потапова Світлана Сергіївна

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:

- Навчитись працювати з файлами, символами, рядковими змінними, бібліотеками.

Теоретичні відомості:

1. Теоретичні відомості з переліком важливих тем:

- Тема №1. Вступ до Роботи з Файлами
- Тема №2. Символи і Рядкові Змінні
- Тема №3. Текстові Файли
- Тема №4. Бінарні Файли
- Тема №5. Стандартна бібліотека та робота з файлами
- Тема №6. Створення й використання бібліотек

2. Індивідуальний план опрацювання теорії:

Тема №1. Вступ до Роботи з Файлами

- Джерела інформації:
 - <https://www.geeksforgeeks.org/basics-file-handling-c/>
- Що опрацьовано:
 - Основні операції з файлами, перевірка стану файлу
- Статус: ознайомлена

Тема №2. Символи і Рядкові Змінні

- Джерела інформації:
 - https://www.w3schools.com/c/c_strings.php
 - https://www.w3schools.com/c/c_strings_functions.php
- Що опрацьовано:
 - Робота з char та string
- Статус: ознайомлена

Тема №3. Текстові Файли

- Джерела інформації:
 - <https://www.geeksforgeeks.org/basics-file-handling-c/>
 - <https://www.geeksforgeeks.org/stdsetbase-stdsetw-stdsetfill-in-cpp/>
- Що опрацьовано:
 - Особливості читання та запису текстових файлів, обробка рядків

- Статус: ознайомлена

Тема №4. Бінарні Файли

- Джерела інформації:
 - <https://www.geeksforgeeks.org/basics-file-handling-c/>
- Що опрацьовано:
 - Робота з бінарними файлами, відмінності від текстових
- Статус: ознайомлена

Тема №5. Стандартна бібліотека та робота з файлами

- Джерела інформації:
 - <https://cplusplus.com/doc/tutorial/files/>
- Що опрацьовано:
 - стандартна бібліотека для роботи з файлами, потоки вводу/виводу: ifstream, ofstream, fstream
- Статус: ознайомлена

Тема №6. Створення й використання бібліотек

- Джерела інформації:
 - <https://www.geeksforgeeks.org/write-header-file-c/>
- Що опрацьовано:
 - Вступ до створення власних бібліотек у C++
- Статус: ознайомлена

Виконання роботи:

1. Опрацювання завдань та вимог до середовища:

Завдання №1. VNS Lab 6 Variant 15

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: Визначити яке слово зустрічається в рядку найчастіше.

Завдання №2. VNS Lab 8 Variant 15

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 "Автомобіль":

- марка;
- рік випуску;
- ціна;
- кольори.

Знищити всі елементи, у яких рік випуску менше заданого, додати елемент на початок файлу.

Завдання №3. VNS Lab 9 Variant 15

Створити текстовий файл F1 не менше, ніж з 10 рядків і записати в нього інформацію

- 1) Скопіювати з файлу F1 у файл F2 всі рядки, що закінчуються на букву «А» і розташовані між рядками з номерами N1 й N2.
- 2) Визначити номер того рядка, у якій найбільше букв «А», файлу F2.

Завдання №4. Algotester Lab 4 Variant 1

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

Вхідні дані

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

Вихідні дані

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

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

Завдання №5. Algotester Lab 6 Variant 1

Вам дано N слів та число K.

Ваше завдання перелічити букви в словах, які зустрічаються в тексті більше-рівне ніж K разів (саме слово, не буква!).

Великі та маленькі букви вважаються однаковими, виводити необхідно малі, посортовані від останньої до першої у алфавіті. Букву потрібно виводити лише один раз.

У випадку якщо таких букв немає - вивести "Empty!".

Вхідні дані

Цілі числа N та K - загальна кількість слів та мінімальна кількість слів щоб враховувати букви цього слова в результаті.

N стрічок s

Вихідні дані

У першому рядку ціле число M - кількість унікальних букв

У другому рядку унікальні букви через пробіли

Завдання №6. Class Practice Task

Задача №1 – Запис текстової стрічки у файл із заданим ім'ям

Реалізувати функцію створення файла 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 файла.

Задача №2 – Копіювання вмісту файла у інший файл

Реалізувати функцію створення файла і запису в нього даних:

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

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

Умови задачі:

- копіювати вміст файла 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 файла.

Завдання №7. Self Practice Task Algotester Lab 4 Variant 2

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

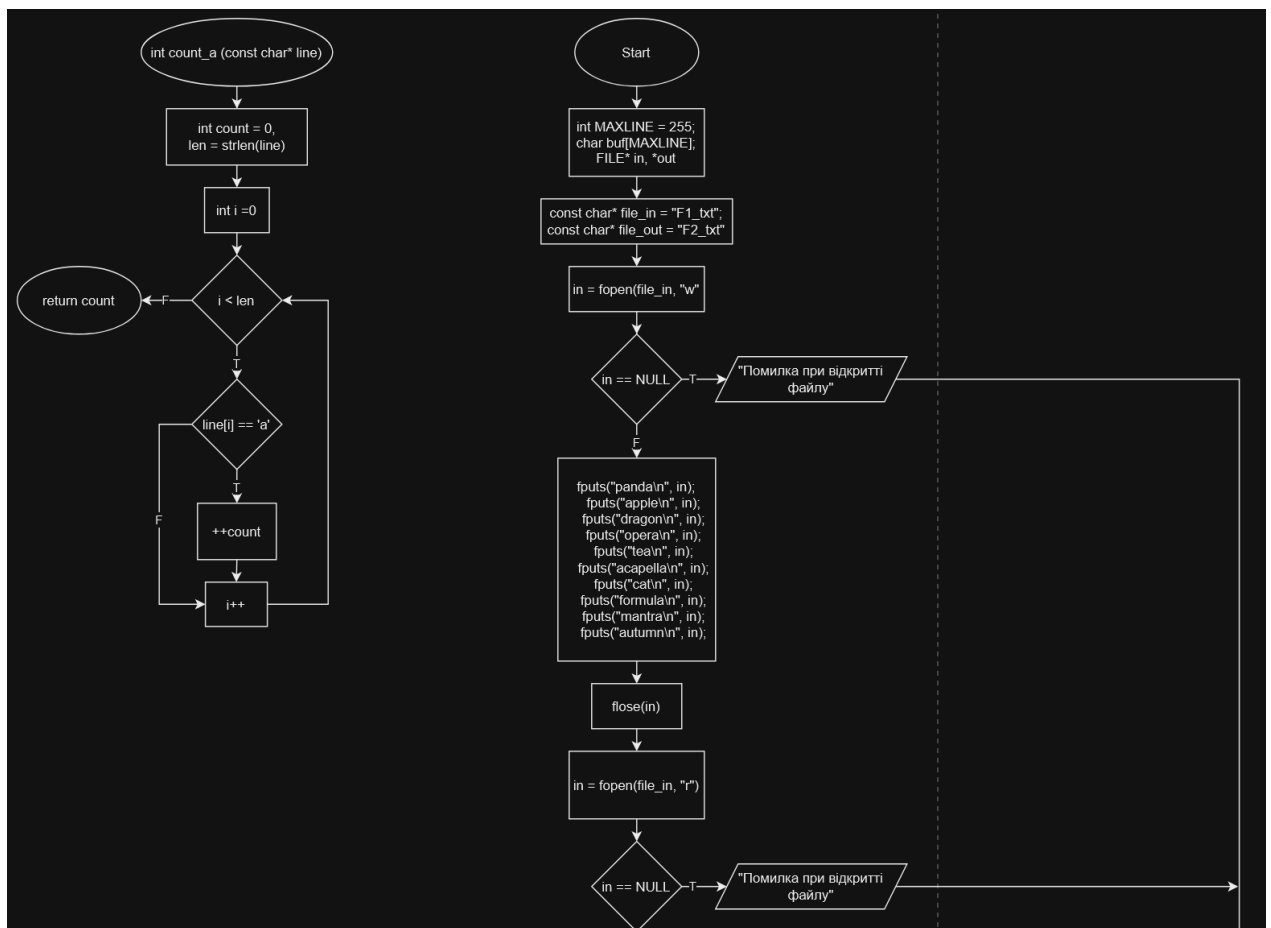
Вихідні дані

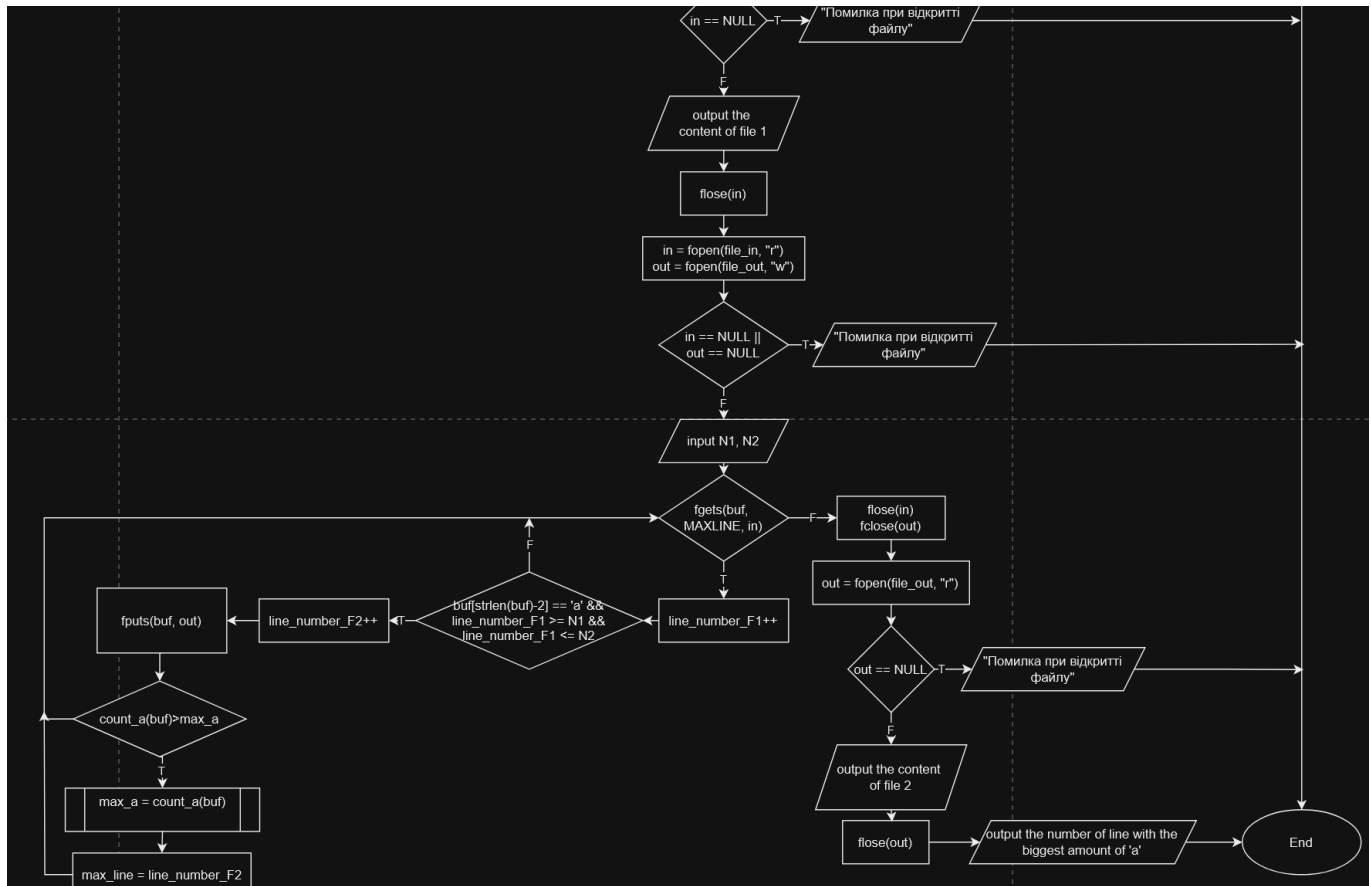
У першому рядку ціле число N - розмір множини a

У наступному рядку N цілих чисел - множина a

2. Дизайн виконання завдань

Завдання №3. VNS Lab 9 Variant 15





3. Код програм з посиланням на зовнішні ресурси

Завдання №1. VNS Lab 6 Variant 15

```

1  #include <iostream>
2  #include <string.h>
3  #include <unordered_map>
4  using namespace std;
5
6  int main() {
7      char str[256];
8      unordered_map<string, int> wordCount;
9
10     cout << "Enter a string: ";
11     gets(str);
12
13     char* token = strtok(str, " ");
14     while (token != NULL) {
15         wordCount[token]++;
16         token = strtok(NULL, " ");
17     }
18
19     int maxCount = 0;
20     char mostFrequentWord[256];
21     for (const auto& el : wordCount) {
22         if (el.second > maxCount) {
23             maxCount = el.second;
24             strcpy(mostFrequentWord, el.first.c_str());
25         }
26     }
27
28     cout << "The word '" << mostFrequentWord << "' is the most frequent" << endl;
29
30     return 0;
31 }

```

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/407/files#diff-07c9536591a1f4f3f5a5227fa455bbe433ebbc39098ddbfe76644ecfc1056907
Завдання №2. VNS Lab 8 Variant 15

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/407/files#diff-4e318e4935a75050a73eca890e5c47f8576bb367bd4a8e9524a35b96d4fb5bc2

```
1  #include <iostream>
2  #include <vector>
3  using namespace std;
4
5  struct Car{
6      char brand[50];
7      int year;
8      int price;
9      char color[50];
10 };
11
12 // функція формування файлу
13 void createFile(const char *filename){
14     FILE* f;
15
16     f = fopen(filename, "wb");
17     if(f==NULL){
18         cout << "Помилка при відкритті файлу";
19         exit(1);
20     }
21
22     int N;
23     cout << "Введіть кількість автомобілей: ";
24     cin >> N;
25
26     Car c;
27
28     for (int i=0; i<N; i++){
29         cout << "Введіть назву марки: ";
30         cin >> c.brand;
31         cout << "Введіть рік випуску: ";
32         cin >> c.year;
33         cout << "Введіть ціну: ";
```



```

34     cin >> c.price;
35     cout << "Введіть колір: ";
36     cin >> c.color;
37
38     fwrite(&c, sizeof(Car), 1, f);
39     if(ferror(f)){
40         cout << "Помилка під час запису в файл";
41         exit(2);
42     }
43 }
44 fclose(f);
45 }
46
47 // функція друку
48 void printFile(const char *filename){
49     FILE* f;
50
51     f = fopen(filename, "rb");
52     if(f==NULL){
53         cout << "Помилка при відкритті файлу";
54         exit(1);
55     }
56
57     Car c;
58
59     cout << endl << "Вмістиме файлу: " << endl << "_____ " << endl;
60
61     while (fread(&c, sizeof(Car), 1, f)) {
62         cout << "Марка: " << c.brand << endl;
63         cout << "Рік випуску: " << c.year << endl;
64         cout << "Ціна: " << c.price << endl;
65
66         cout << "Колір: " << c.color << endl;
67         cout << "_____ " << endl;
68     }
69     fclose(f);
70
71 // функція додвання
72 void addNewCar(const char *filename){
73     FILE* f;
74
75     f = fopen(filename, "rb");
76     if(f==NULL){
77         cout << "Помилка при відкритті файлу";
78         exit(2);
79     }
80
81     vector<Car> cars;
82     Car c;
83
84     while (fread(&c, sizeof(Car), 1, f)){
85         cars.push_back(c);
86     }
87
88     fclose(f);
89
90     f = fopen(filename, "wb");
91     if(f==NULL){
92         cout << "Помилка при відкритті файлу";
93         exit(3);
94     }
95

```

```

96     cout << "Введіть дані нового автомобіля: " << endl << "Введіть назву марки: ";
97     cin >> c.brand;
98     cout << "Введіть рік випуску: ";
99     cin >> c.year;
100    cout << "Введіть ціну: ";
101    cin >> c.price;
102    cout << "Введіть колір: ";
103    cin >> c.color;
104
105    fwrite(&c, sizeof(Car), 1, f);
106    if(ferror(f)){
107        cout << "Помилка під час запису \u0444айл";
108        exit(4);
109    }
110
111    for (const auto& car : cars) {
112        fwrite(&car, sizeof(Car), 1, f);
113    }
114    if(ferror(f)){
115        cout << "Помилка під час запису \u0444айл";
116        exit(5);
117    }
118    fclose(f);
119 }
120
121 // функція знищення
122 void deleteCar(const char *filename, int yr){
123     FILE* f;
124
125     f = fopen(filename, "rb");
126     if(f==NULL){
127         cout << "Помилка при відкритті файлу";
128         exit(1);

```

```

129     }
130
131     vector<Car> cars;
132     Car c;
133
134     while (fread(&c, sizeof(Car), 1, f)){
135         cars.push_back(c);
136     }
137     fclose(f);
138
139     vector<Car> filteredByYear;
140     for(const auto& car : cars){
141         if(car.year >= yr){
142             filteredByYear.push_back(car);
143         }
144     }
145
146     f = fopen(filename, "wb");
147     if(f==NULL){
148         cout << "Помилка при відкритті файлу";
149         exit(2);
150     }
151
152     for (const auto& car : filteredByYear) {
153         fwrite(&car, sizeof(Car), 1, f);
154     }
155
156     if(ferror(f)){
157         cout << "Помилка під час запису \u0444айл";
158         exit(3);
159     }

```

```

160
161     fclose(f);
162 }
163
164 int main(){
165     const char* filename = "cars.dat";
166     int yr;
167
168     createFile(filename);
169     printFile(filename);
170     addNewCar(filename);
171     printFile(filename);
172     cout << "Вкажіть рік: ";
173     cin >> yr;
174     cout << endl << "Знищено автомобілі, рік випуску яких менше заданого.";
175     deleteCar(filename, yr);
176     printFile(filename);
177
178     return 0;
179 }

```

Завдання №3. VNS Lab 9 Variant 15

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/407/files#diff-0e8a95e658593441b6dd292b40a97c885f111ad0c0265bd02afbeb2fd9bcc5ce

```

1  #include <iostream>
2  #include <string.h>
3  using namespace std;
4
5  int count_a (const char* line){
6      int count = 0;
7      int len = strlen(line);
8      for (int i=0; i<len; i++) {
9          if (line[i] == 'a') {
10             ++count;
11         }
12     }
13     return count;
14 }
15
16 int main(){
17     int MAXLINE=255;
18     char buf[MAXLINE];
19     FILE *in, *out;
20
21     const char* file_in = "F1.txt";
22     const char* file_out = "F2.txt";
23
24     in = fopen(file_in, "w");
25     if(in==NULL){
26         cout << "Помилка при відкритті файлу";
27         exit(1);
28     }
29
30     fputs("panda\n", in);
31     fputs("apple\n", in);
32     fputs("dragon\n", in);
33     fputs("opera\n", in);

```

```

34     fputs("tea\n", in);
35     fputs("acapella\n", in);
36     fputs("cat\n", in);
37     fputs("formula\n", in);
38     fputs("mantra\n", in);
39     fputs("autumn\n", in);
40     fclose(in);
41
42     in = fopen(file_in, "r");
43
44     if (in==NULL){
45         cout << "Помилка при відкритті файлу";
46         exit(1);
47     }
48
49     cout << "Вміст файлу F1:" << endl;
50     while (fgets(buf, MAXLINE, in)) {
51         cout << buf;
52     }
53     fclose(in);
54
55     in = fopen(file_in, "r");
56     out = fopen(file_out, "w");
57
58     if (in==NULL || out==NULL){
59         cout << "Помилка при відкритті файлу";
60         exit(1);
61     }
62
63     int max_a = 0, max_line, line_number_F1 = 0, line_number_F2 = 0, N1, N2;
64     cout << "Введіть N1 та N2: ";
65     cin >> N1 >> N2;
66
67     while(fgets(buf, MAXLINE, in)){
68         line_number_F1++;
69         if (buf[strlen(buf)-2] == 'a' && line_number_F1 >= N1 && line_number_F1 <= N2){
70             line_number_F2++;
71             fputs(buf, out);
72             if(count_a(buf)>max_a){
73                 max_a = count_a(buf);
74                 max_line = line_number_F2;
75             }
76         }
77     }
78
79     fclose(in);
80     fclose(out);
81
82     out = fopen(file_out, "r");
83     if(out==NULL){
84         cout << "Помилка при відкритті файлу";
85         exit(1);
86     }
87
88     cout << "Вміст вихідного файлу: " << endl;
89     while (fgets(buf, MAXLINE, out)) {
90         cout << buf;
91     }
92
93     fclose(out);
94     cout << "Найбільше букв 'a' містить рядок №" << max_line;
95     return 0;
96 }

```

Завдання №4. Algotester Lab 4 Variant 1

1) https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/407/files#diff-4f15bc05e03136535f2cd7aff84e8215d468c72d63715363e8efd01c76bdd1e8

```
1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4  #include <set>
5
6  using namespace std;
7
8  void printRes(vector<int>& array){
9      cout << array.size() << endl;
10     for (int num : array){
11         cout << num << " ";
12     }
13     cout << endl << endl;
14 }
15
16 int main(){
17     int N, M;
18     cin >> N;
19     vector<int> arr1(N);
20
21     for(int i=0; i<N; i++){
22         cin >> arr1[i];
23     }
24
25     cin >> M;
26     vector<int> arr2(M);
27
28     for(int i=0; i<M; i++){
29         cin >> arr2[i];
30     }
31
32     sort(arr1.begin(), arr1.end());
33     sort(arr2.begin(), arr2.end());
```

```
34
35     vector<int> N_minus_M;
36     set_difference(arr1.begin(), arr1.end(), arr2.begin(), arr2.end(), back_inserter(N_minus_M));
37
38     vector<int> M_minus_N;
39     set_difference(arr2.begin(), arr2.end(), arr1.begin(), arr1.end(), back_inserter(M_minus_N));
40
41     vector<int> intersection;
42     set_intersection(arr1.begin(), arr1.end(), arr2.begin(), arr2.end(), back_inserter(intersection));
43
44     vector<int> unionres;
45     set_union(arr1.begin(), arr1.end(), arr2.begin(), arr2.end(), back_inserter(unionres));
46
47     vector<int> sym_diff;
48     set_symmetric_difference(arr1.begin(), arr1.end(), arr2.begin(), arr2.end(), back_inserter(sym_diff));
49
50     printRes(N_minus_M);
51     printRes(M_minus_N);
52     printRes(intersection);
53     printRes(unionres);
54     printRes(sym_diff);
55
56     return 0;
57 }
```

2) https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/407/files#diff-92cbe5f8c9185bfd68392b1717c6cb03a07a1552f413fa360f81384708fa7186

```
1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4  using namespace std;
5
6  void printRes(vector<int>& array){
7      sort(array.begin(), array.end());
8      cout << array.size() << endl;
9      for (int num : array){
10         cout << num << " ";
11     }
12     cout << endl << endl;
13 }
14
15 bool found(const vector<int>& vec, int num){
16     for (int n : vec) {
17         if (n == num) {
18             return true;
19         }
20     }
21     return false;
22 }
23
24 int main(){
25     int N, M;
26     cin >> N;
27     vector<int> arr1(N);
28
29     for(int i=0; i<N; i++){
30         cin >> arr1[i];
31     }
32
33     cin >> M;
34
35     vector<int> arr2(M);
36
37     for(int i=0; i<M; i++){
38         cin >> arr2[i];
39     }
40
41     // N - M
42     vector<int> N_minus_M;
43     for (int num : arr1) {
44         if (!found(arr2, num)) {
45             N_minus_M.push_back(num);
46         }
47     }
48
49     //M - N
50     vector<int> M_minus_N;
51     for (int num : arr2) {
52         if (!found(arr1, num)) {
53             M_minus_N.push_back(num);
54         }
55     }
56
57     // Перетин
58     vector<int> intersection;
59     for (int num : arr1) {
60         if (found(arr2, num)) {
61             intersection.push_back(num);
62         }
63     }
64
65     // Объединения
66     vector<int> union_set;
```

```

65     vector<int> union_set;
66     for (int num : arr1) {
67         if (!found(union_set, num)) {
68             union_set.push_back(num);
69         }
70     }
71
72     for (int num : arr2) {
73         if (!found(union_set, num)) {
74             union_set.push_back(num);
75         }
76     }
77
78     // Симетрична різниця
79     vector<int> sym_diff;
80     for (int num : arr1) {
81         if (!found(arr2, num)) {
82             sym_diff.push_back(num);
83         }
84     }
85
86     for (int num : arr2) {
87         if (!found(arr1, num)) {
88             sym_diff.push_back(num);
89         }
90     }
91     printRes(N_minus_M);
92     printRes(M_minus_N);
93     printRes(intersection);
94     printRes(union_set);
95     printRes(sym_diff);
96     return 0;
97 }

```

Завдання №5. Algotester Lab 6 Variant 1

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/407/files#diff-155662c1c850d4f3ea5b52e6912fd1fd157810768a13d09bb4b2d903453eaa5b

```

1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4  #include <unordered_map>
5  #include <set>
6  using namespace std;
7
8  int main() {
9      int n, k;
10     cin >> n >> k;
11     cin.ignore();
12
13     vector<string> w(n);
14
15     for (int i = 0; i < n; i++) {
16         getline(cin, w[i]);
17         for (char &c : w[i]) {
18             c = tolower(c);
19         }
20     }
21
22     unordered_map<string, int> word_count;
23
24     for (const string& word : w) {
25         word_count[word]++;
26     }
27
28     set<char> result_set;
29
30     for (const auto& entry : word_count) {
31         if (entry.second >= k) {
32             for (char c : entry.first) {
33                 result_set.insert(c);
34             }
35         }
36     }
37
38     for (char c : result_set) {
39         cout << c << " ";
40     }
41     cout << endl;
42     return 0;
43 }

```

```

34         }
35     }
36 }
37
38 if (result_set.empty()) {
39     cout << "Empty!";
40     return 0;
41 }
42
43 vector<char> result(result_set.begin(), result_set.end());
44
45 sort(result.rbegin(), result.rend());
46
47 cout << result.size() << endl;
48
49 for (const char &c : result) {
50     cout << c << " ";
51 }
52
53 return 0;
54 }

```

Завдання №6. Class Practice Task

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/407/files#diff-5397da9e4a29860b5cd20eac142db7fa3f878e5fbc193dd84da028b9b3534144

```

1  ✓ #include <iostream>
2  #include <fstream>
3  using namespace std;
4
5  enum FileOpResult {Success, Failure_create, Failure_open, Failure_write};
6  ✓ FileOpResult write_to_file(char *name, char *content){
7
8  ✓   if (name == nullptr){
9       return Failure_create;
10   }
11
12   ofstream file(name);
13  ✓   if (!file.is_open()) {
14       return Failure_open;
15   }
16
17   file << content;
18
19  ✓   if(file.fail()){
20       return Failure_write;
21   }
22
23   file.close();
24   return Success;
25 }
26
27  ✓ int main(){
28     char filename[124];
29     char content[1024];
30
31     cout << "Enter filename: ";
32     cin.getline(filename, 100);
33

```


2) https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/407/files#diff-93a8d3821cc1465c4f29075e4920f622ca7d17d39dff7fa5aef9a9bbecff5746

```
1  #include <iostream>
2  #include <fstream>
3  using namespace std;
4
5  enum FileOpResult {Success, Failure_create, Failure_open, Failure_write};
6  FileOpResult copy_file(const char *file_from, const char *file_to){
7
8      ifstream file1(file_from);
9      if (!file1.is_open()){
10         return Failure_open;
11     }
12
13     ofstream file2(file_to);
14     if (!file2.is_open()){
15         cerr << "2";
16         return Failure_create;
17     }
18
19     file2 << file1.rdbuf();
20     if (file2.fail()){
21         cerr << "3";
22         return Failure_write;
23     }
24
25     file1.close();
26     file2.close();
27
28     return Success;
29 }
30
31 int main(){
32     const char* filename1 = "file_from.txt";
33     const char* filename2 = "file_to.txt";
34
35     FileOpResult res = copy_file(filename1, filename2);
36
37     if(res == Success){
38         cout << "Копіювання даних у файл пройшло успішно";
39     }
40     else if(res == Failure_create){
41         cout << "Не вдалося відкрити або створити вихідний файл";
42     }
43     else if(res == Failure_open){
44         cout << "Не вдалося відкрити вхідний файл";
45     }
46     else if(res == Failure_write){
47         cout << "Помилка під час запису даних у файл";
48     }
49
50     return 0;
51 }
```

Завдання №7. Self Practice Task Algotester Lab 4 Variant 2

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/407/files#diff-

dda290cf99115fe4b0ab935b096ff7b75bb2b3265a9f77fe222c103350728e75

```
1  #include <iostream>
2  #include <vector>
3  using namespace std;
4
5  void removeDuplicates(vector<int>& array) {
6      for (int i = 0; i < array.size(); ++i) {
7          for (int j = i + 1; j < array.size(); ++j) {
8              if (array[i] == array[j]) {
9                  array.erase(array.begin() + j);
10                 --j;
11             }
12         }
13     }
14 }
15
16 void bubbleSort(vector<int>& arr){
17     for (int i = 0; i < arr.size() - 1; ++i) {
18         for (int j = i + 1; j < arr.size(); ++j) {
19             if (arr[i] > arr[j]) {
20                 swap(arr[i], arr[j]);
21             }
22         }
23     }
24 }
25
26 void rotateArray(vector<int>& arr, int K) {
27     K = K % arr.size();
28     vector<int> temp1(arr.begin(), arr.begin() + K);
29     vector<int> temp2(arr.begin() + K, arr.end());
30     arr = temp2;
31     arr.insert(arr.end(), temp1.begin(), temp1.end());
32 }
33
34 int main(){
35     int N, K;
36     cin >> N >> K;
37     vector<int> arr(N);
38     for (int i = 0; i < N; ++i) {
39         cin >> arr[i];
40     }
41
42     removeDuplicates(arr);
43     bubbleSort(arr);
44     rotateArray(arr, K);
45
46     cout << arr.size() << endl;
47     for (int num : arr) {
48         cout << num << " ";
49     }
50
51     return 0;
52 }
```

4. Результати виконання завдань та фактично затрачений час Завдання №1. VNS Lab 6 Variant 15

```
Enter a string: cat dog dog man dog
The word 'dog' is the most frequent
```

Планований час: 40 хв, фактично: 40 хв

Завдання №2. VNS Lab 8 Variant 15

Введіть кількість автомобілей: 2
Введіть назву марки: BMW
Введіть рік випуску: 2005
Введіть ціну: 5000
Введіть колір: black
Введіть назву марки: Audi
Введіть рік випуску: 2020
Введіть ціну: 50000
Введіть колір: black

Вмістиме файлу:

Марка: BMW
Рік випуску: 2005
Ціна: 5000
Колір: black

Марка: Audi
Рік випуску: 2020
Ціна: 50000
Колір: black

Введіть дані нового автомобіля:
Введіть назву марки: Tesla
Введіть рік випуску: 2024
Введіть ціну: 45000
Введіть колір: gray

Вмістиме файлу:

Марка: Tesla
Рік випуску: 2024
Ціна: 45000
Колір: gray

Марка: BMW
Рік випуску: 2005
Ціна: 5000
Колір: black

Марка: Audi
Рік випуску: 2020
Ціна: 50000
Колір: black

Вкажіть рік: 2007

Знищено автомобілі, рік випуску яких менше заданого.
Вмістиме файлу:

Марка: Tesla
Рік випуску: 2024
Ціна: 45000
Колір: gray

Марка: Audi
Рік випуску: 2020
Ціна: 50000
Колір: black

Планований час: 2 год, фактично: 1,5 год

Завдання №3. VNS Lab 9 Variant 15

```
Вміст файлу F1:
panda
apple
dragon
opera
tea
acapella
cat
formula
mantra
autumn
Введіть N1 та N2: 2 8
Вміст вихідного файлу:
opera
tea
acapella
formula
Найбільше букв 'а' містить рядок №3
```

Планований час: 1,5 год, фактично: 1,5 год

Завдання №4. Algotester Lab 4 Variant 1

```
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
```

Планований час: 40 хв, фактично: 1,5 год

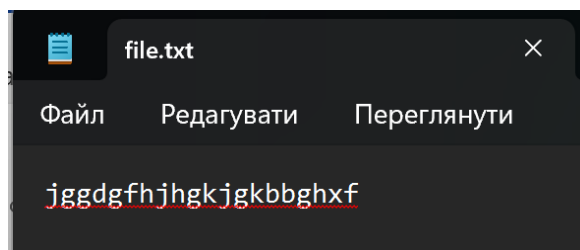
Завдання №5. Algotester Lab 6 Variant 1

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

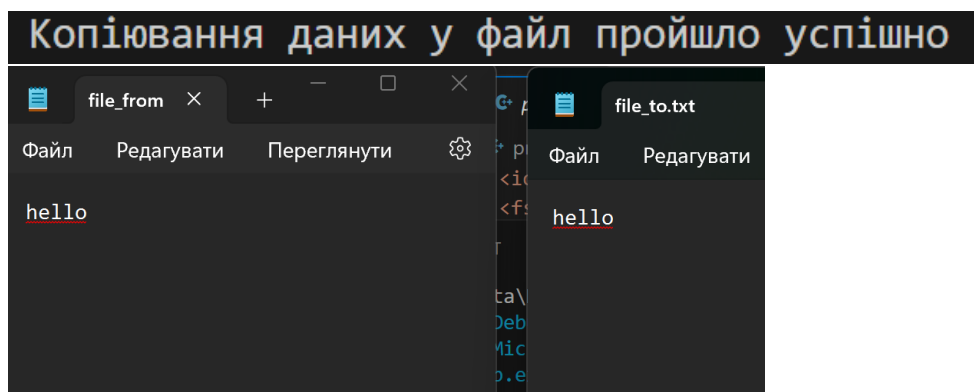
Планований час: 40 хв, фактично: 40 хв

Завдання №6. Class Practice Task

```
Enter filename: file.txt
Enter content: jggdgfhjhgkjgkbbghxf
Запис даних у файл пройшов успішно
```



Планований час: 50 хв, фактично: 50 хв



Планований час: 40 хв, фактично: 50 хв

Завдання №7. Self Practice Task Algotester Lab 4 Variant 2

```
7 3
1 2 2 3 3 4 5
5
4 5 1 2 3
```

Планований час: 30 хв, фактично: 30 хв

5. Кооперація з командою

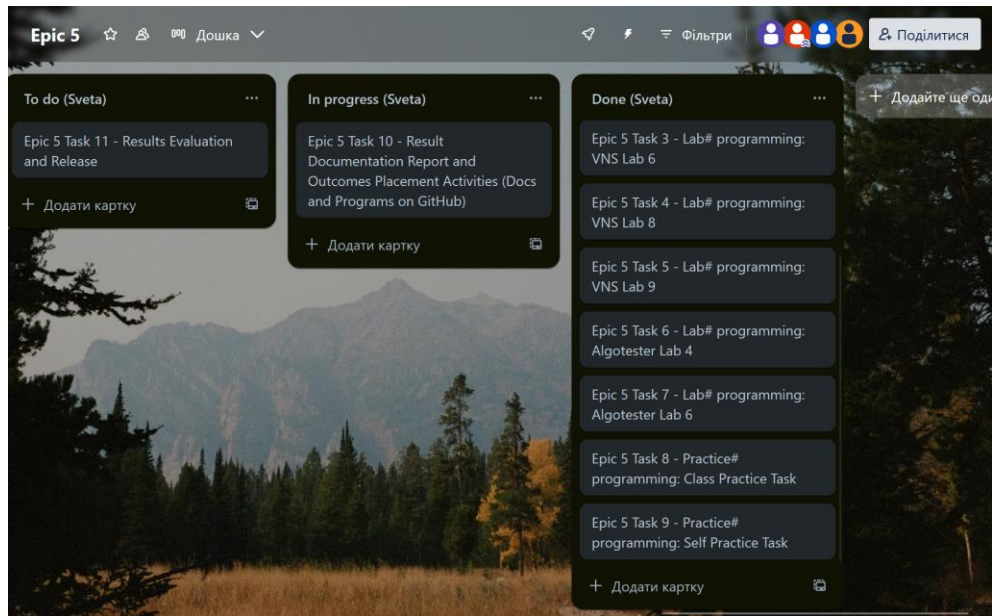


Рисунок 1. Trello



Висновок: Під час виконання роботи я теоретично ознайомилась та на практиці закріпила знання про роботу з текстовими, бінарними файлами у мові C/C++.