

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-13

Литвин Маркіян Назарович

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:** Навчитися працювати з файлами, символами і рядковими змінними та текстовими файлами, стандартною бібліотекою.

### **Теоретичні відомості:**

- файли
- рядкові змінні та символи
- бібліотеки

### **Джерела:**

- <https://www.youtube.com/watch?v=FeNqHytI0fA>
- ChatGPT
- <https://www.youtube.com/watch?app=desktop&v=SSNJ7alki-E>
- <https://itproger.com/ua/course/cpp/13>

## **Виконання роботи**

### **Завдання 1: VNS Lab 6 - Task 1-17**

#### **Умова:**

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.

17. Всі слова рядка, які починаються з букви, відсортувати за абеткою.

#### **Розв'язок:**

```

1  #include <iostream>
2  #include <string>
3  #include <stdio.h>
4  #include <string.h>
5  #include <vector>
6  #include <algorithm>
7  using namespace std;
8
9  void sort_words(char s[]){
10     vector<string> words;
11     char* token = strtok(s, " ");
12     while (token != NULL) {
13         string word(token);
14
15         if (!word.empty() && isalpha(word[0])) {
16             words.push_back(word);
17         }
18         token = strtok(NULL, " ");
19     }
20     sort(words.begin(), words.end());
21     cout << "Words that started with a letter:" << endl;
22     for (const auto& word : words) {
23         cout << word << endl;
24     }
25 }
26 int main() {
27     cout << "Enter words: ";
28     char s[256];
29     gets(s);
30     sort_words(s);
31
32     return 0;
33 }
34

```

**Результат:**

```

Enter words: epic 5 nulp ai markiian
Words that started with a letter:
ai
epic
markiian
nulp
PS D:\Epics>

```

Час виконання ~ 35 хв

## Завдання 2: VNS Lab 8 - Task 1-17

### Умова:

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.

#### 17. Структура "Фільм":

- назва;
- режисер;
- рік випуску;
- вартість.

Знищити всі елементи, у яких вартість перевищує задану, додати елемент на початок файлу.

## Розв'язок:

```
1 #include <iostream>
2 #include <cstdio>
3 #include <vector>
4 #include <string>
5
6 using namespace std;
7
8 struct Film {
9     char name[100];
10    char director[100];
11    int year;
12    int cost;
13 };
14
15 const char* filename = "film.dat";
16
17 void create_file() {
18     FILE* file = fopen(filename, "wb");
19     if (!file) {
20         cerr << "Error." << endl;
21         return;
22     }
23     while (true) {
24         Film film;
25         cout << "Enter film name (or press Enter to stop): ";
26         cin.ignore();
27         cin.getline(film.name, 100);
28
29         if (strlen(film.name) == 0) {
30             break;
31         }
32         cout << "Director: ";
33         cin.getline(film.director, 100);
34         cout << "Year: ";
35         cin >> film.year;
36         cout << "Cost in millions: ";
37         cin >> film.cost;
38         fwrite(&film, sizeof(Film), 1, file);
39     }
40     fclose(file);
41 }
42
43 void create_file(const vector<Film>& films) {
44     FILE* file = fopen(filename, "wb");
45     if (!file) {
46         cerr << "Error." << endl;
47         return;
48     }
49     for (const auto& film : films) {
50         fwrite(&film, sizeof(Film), 1, file);
51     }
52     fclose(file);
53 }
54
55 void print_file() {
56     FILE* file = fopen(filename, "rb");
57     if (!file) {
58         cerr << "Error." << endl;
59         return;
60     }
61     Film film;
62     while (fread(&film, sizeof(Film), 1, file)) {
63         {
64             cout << "Name: " << film.name
65                 << ", Director: " << film.director
66                 << ", Year: " << film.year
67                 << ", Cost in millions: " << film.cost << "\n";
68         }
69         fclose(file);
70     }
71 }
72
73 void delete_expensive(double max_price) {
74     FILE* file = fopen(filename, "rb");
75     if (!file) {
76         cerr << "Error." << endl;
77         return;
78     }
79     vector<Film> films;
80     Film film;
81
82     while (fread(&film, sizeof(Film), 1, file)) {
83         if (film.cost <= max_price) {
84             films.push_back(film);
85         }
86     }
87     fclose(file);
88
89     file = fopen(filename, "wb");
90     if (!file) {
91         cerr << "Error." << endl;
92         return;
93     }
94
95     for (const auto& f : films) {
96         fwrite(&f, sizeof(Film), 1, file);
97     }
98     fclose(file);
99 }
100
101 void add_file() {
102     FILE* new_file;
103     cout << "Enter file name (or press Enter to stop): " << endl;
104     cout << "Name: "; cin.ignore(); cin.getline(new_file.name, 100);
105     if (strlen(new_file.name) == 0) {
106         return;
107     }
108     cout << "Director: "; cin.getline(new_file.director, 100);
109     cout << "Year: "; cin >> new_file.year;
110     cout << "Cost: "; cin >> new_file.cost;
111     FILE* file = fopen(filename, "rb");
112     vector<Film> films;
113     Film film;
114     while (fread(&film, sizeof(Film), 1, file)) {
115         films.push_back(film);
116     }
117     fclose(file);
118
119     file = fopen(filename, "wb");
120     fwrite(new_file, sizeof(Film), 1, file);
121     for (const auto& f : films) {
122         fwrite(&f, sizeof(Film), 1, file);
123     }
124     fclose(file);
125 }
126
127 int main() {
128     vector<Film> primary_films = {
129         {"One piece", "Eiichiro Oda", 1999, 150},
130         {"Home Alone", "Chris Columbus", 1990, 120},
131         {"Interstellar", "Christopher Nolan", 2009, 180},
132         {"Spider-Man: No Way Home", "John Watts", 1999, 150},
133         {"Right Club", "David Fincher", 1999, 170},
134         {"Titanic", "James Cameron", 1997, 150},
135         {"Harry Potter and the Half-Blood Prince", "David Yt", 2009, 240}
136     };
137     create_file(primary_films);
138 }
```

```

140     create_file(primary_files);
141
142     cout << "File's content:" << endl;
143     print_file();
144     int max_price;
145     cout << "\nMax cost: ";
146     cin >> max_price;
147     delete_expensive(max_price);
148     cout << "\nFile after removing films:" << endl;
149     print_file();
150     add_film();
151     cout << "\nFile after new films added:" << endl;
152     print_file();
153
154     return 0;
155 }
156

```

## Результат:

```

File's content:
Name: One piece, Director: Eiichiro Oda, Year: 1999, Cost in millions: 150
Name: Home Alone, Director: Chris Columbus, Year: 1990, Cost in millions: 120
Name: Interstellar, Director: Christoper Nolan, Year: 2009, Cost in millions: 310
Name: Spider Man: No Way Home, Director: John Watts, Year: 1999, Cost in millions:
350
Name: Fight Club, Director: David Fincer, Year: 1999, Cost in millions: 170
Name: Titanic, Director: James Cameron, Year: 1997, Cost in millions: 190
Name: Harry Potter and the Half-Blood Prince, Director: David Ets, Year: 2009, Cos
t in millions: 240

Max cost: 250

File after removing films:
Name: One piece, Director: Eiichiro Oda, Year: 1999, Cost in millions: 150
Name: Home Alone, Director: Chris Columbus, Year: 1990, Cost in millions: 120
Name: Fight Club, Director: David Fincer, Year: 1999, Cost in millions: 170
Name: Titanic, Director: James Cameron, Year: 1997, Cost in millions: 190
Name: Harry Potter and the Half-Blood Prince, Director: David Ets, Year: 2009, Cos
t in millions: 240
Enter film name (or press Enter to stop):
Name: Venom
Director: Marik
Year: 2024
Cost: 260

File after new films added:
Name: Venom, Director: Marik, Year: 2024, Cost in millions: 260
Name: One piece, Director: Eiichiro Oda, Year: 1999, Cost in millions: 150
Name: Home Alone, Director: Chris Columbus, Year: 1990, Cost in millions: 120
Name: Fight Club, Director: David Fincer, Year: 1999, Cost in millions: 170
Name: Titanic, Director: James Cameron, Year: 1997, Cost in millions: 190
Name: Harry Potter and the Half-Blood Prince, Director: David Ets, Year: 2009, Cos
t in millions: 240
PS D:\Enics>

```

Час виконання ~ 1 год

## Завдання 3: VNS Lab 9 - Task 1-17

### Умова:

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

### Розв'язок:

```

1  #include <iostream>
2  #include <fstream>
3  #include <string>
4  #include <vector>
5  #include <cctype>
6  using namespace std;
7
8  void file_F1(const string& filename) {
9      ofstream file(filename);
10     cout << "Enter content for F1:\n";
11     string line;
12     while (true) {
13         getline(cin, line);
14         if (line.empty()) break;
15         file << line << "\n";
16     }
17     file.close();
18 }
19
20 vector<string> read(const string& filename) {
21     ifstream file(filename);
22     vector<string> lines;
23     string line;
24     while (getline(file, line)) {
25         lines.push_back(line);
26     }
27     file.close();
28     return lines;
29 }
30 void write(const string& filename, const vector<string>& lines) {
31     ofstream file(filename);
32     for (const auto& line : lines) {
33         file << line << "\n";
34     }
35     file.close();
36 }
37 vector<string> filter(const vector<string>& lines, int N1, int N2) {
38     vector<string> filtered_lines;
39     for (int i = N1 - 1; i < N2; ++i) {
40         if (!lines[i].empty() && lines[i][0] == 'A') {
41             filtered_lines.push_back(lines[i]);
42         }
43     }
44     for (int i = N2 + 2; i < lines.size(); ++i) {
45         filtered_lines.push_back(lines[i]);
46     }
47     return filtered_lines;
48 }
49 int count(const string& line) {
50     int wordCount = 0;
51     bool inWord = false;
52     for (char c : line) {
53         if (isspace(c)) {
54             if (inWord) {
55                 ++wordCount;
56                 inWord = false;
57             }
58         } else {
59             inWord = true;
60         }
61     }
62     if (inWord) {
63         ++wordCount;
64     }
65     return wordCount;
66 }
67 int main() {
68     const string F1 = "F1.txt";
69     const string F2 = "F2.txt";
70
71     file_F1(F1);
72     int N1, N2;
73     cout << "Введіть N1 та N2: ";
74     cin >> N1 >> N2;
75     cin.ignore();
76     vector<string> linesF1 = read(F1);
77     vector<string> linesF2 = filter(linesF1, N1, N2);
78     write(F2, linesF2);
79
80     if (!linesF2.empty()) {
81         const string& lastLine = linesF2.back();
82         int wordCount = count(lastLine);
83         cout << "Кількість слів в останньому рядку F2: " << wordCount << endl;
84     } else {
85         cout << "Файл F2 порожній." << endl;
86     }
87     return 0;
88 }
89

```

## Результат:

```
Enter content for F1:
aad sadkjdkasj dsaklj
AS SAKajsdkj adkljaskldj
AS KSAJKJS
sASAKJ
AS AKJSAKJS
ASKAJSKAJ SAKJSAKJS
ASKAJkjs asjl

Введіть N1 та N2: 1 3
Кількість слів в останньому рядку F2: 2
PS D:\Epics>

pic5 > F1.txt
1 aad sadkjdkasj dsaklj
2 AS SAKajsdkj adkljaskldj
3 AS KSAJKJS
4 sASAKJ
5 AS AKJSAKJS
6 ASKAJSKAJ SAKJSAKJS
7 ASKAJkjs asjl
8

1 ASKAJSKAJ SAKJSAKJS
2 ASKAJkjs asjl
3
```

Час виконання ~ 1.5 год

## Завдання 4: Algotester Lab4v1

### Умова:

Вам дано 2 цілих чисел масиви, розміром NN та MM.  
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 NN - розмір масиву 1  
У другому рядку NN цілих чисел - елементи масиву 1  
У третьому рядку ціле число MM - розмір масиву 2  
У четвертому рядку MM цілих чисел - елементи масиву 2

### Output

Вивести результат виконання 5 вищезазначених операцій у форматі:

У першому рядку ціле число NN - розмір множини  
У наступному рядку NN цілих чисел - посортована у порядку зростання множина

### Розв'язок:

```

1  #include <iostream>
2  #include <vector>
3  #include <set>
4  #include <algorithm>
5  using namespace std;
6
7  void Print(const vector<int>& result) {
8      cout << result.size() << endl;
9      for(int elm : result) {
10         cout << elm << " ";
11     }
12     cout << endl << endl;
13 }
14
15 int main() {
16     int N, M;
17     cin >> N;
18     vector<int> arr1(N);
19     for(int i = 0; i < N; i++) {
20         cin >> arr1[i];
21     }
22
23     cin >> M;
24     vector<int> arr2(M);
25     for(int i = 0; i < M; i++) {
26         cin >> arr2[i];
27     }
28
29     sort(arr1.begin(), arr1.end());
30     sort(arr2.begin(), arr2.end());
31
32     vector<int> result;
33
34     set_difference(arr1.begin(), arr1.end(), arr2.begin(), arr2.end(), back_inserter(result));
35     Print(result);
36     result.clear();
37
38     set_difference(arr2.begin(), arr2.end(), arr1.begin(), arr1.end(), back_inserter(result));
39     Print(result);
40     result.clear();
41
42     set_intersection(arr1.begin(), arr1.end(), arr2.begin(), arr2.end(), back_inserter(result));
43     Print(result);
44     result.clear();
45
46     set_union(arr1.begin(), arr1.end(), arr2.begin(), arr2.end(), back_inserter(result));
47     Print(result);
48     result.clear();
49
50     set_symmetric_difference(arr1.begin(), arr1.end(), arr2.begin(), arr2.end(), back_inserter(result));
51     Print(result);
52     result.clear();
53
54     return 0;
55 }
56

```

**Результат:**

```

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

3
6 7 8

1
5

8
1 2 3 4 5 6 7 8

7
1 2 3 4 6 7 8

PS D:\Epics>

```

Created	Compiler	Result	Time (sec.)	Memory (MiB)	Actions
21 hours ago	C++ 23	Accepted	0.003	1.262	<a href="#">View</a>

Час виконання ~ 35 хв

**Завдання 5: Algotester Lab4v2**



## Умова:

Вам дано масив  $aa$  з  $NN$  цілих чисел.

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

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

## Input

У першому рядку цілі числа  $NN$  та  $KK$

У другому рядку  $NN$  цілих чисел - елементи масиву  $aa$

## Output

У першому рядку ціле число  $NN$  - розмір множини  $aa$

У наступному рядку  $NN$  цілих чисел - множина  $a$

## Розв'язок:

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

## Результат:

```
10 3
1 2 2 3 3 3 4 5 6 7
7
4 5 6 7 1 2 3
PS D:\Epics> |
```

Created	Compiler	Result	Time (sec.)	Memory (MiB)	Actions
8 hours ago	C++ 23	Accepted	0.003	1.223	<a href="#">View</a>

Час виконання ~ 25 хв

## Завдання 6: VNS Lab 6v1

### Умова:

Вам дано NN слів та число KK.

Ваше завдання перерахувати букви в словах, які зустрічаються в тексті більше-рівне ніж KK 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!".

### Input

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

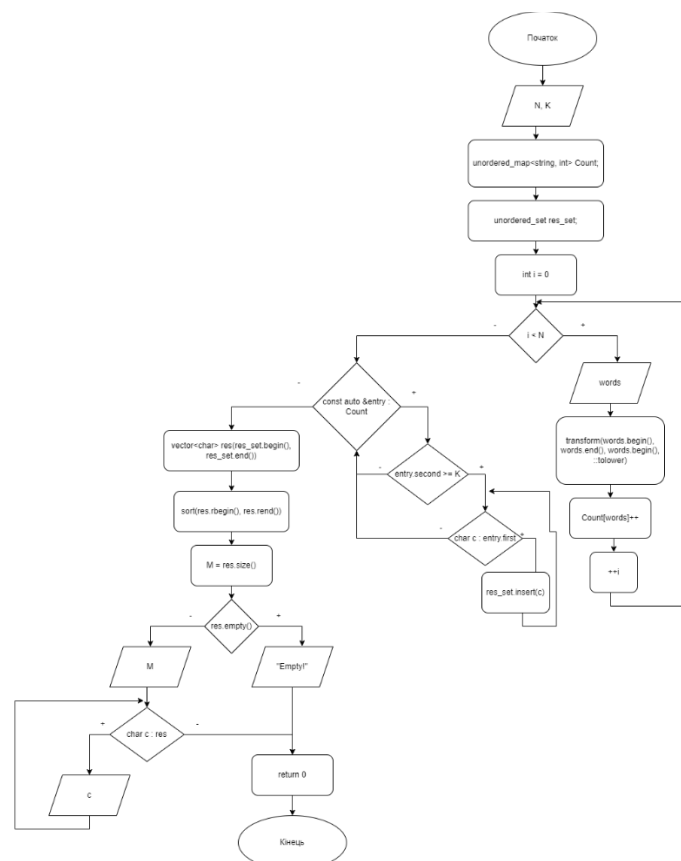
NN стрічок ss

### Output

У першому рядку ціле число MM - кількість унікальних букв

У другому рядку унікальні букви через пробіли

### Блок-схема:



## Розв'язок:

```
1  #include <iostream>
2  #include <vector>
3  #include <unordered_map>
4  #include <set>
5  #include <unordered_set>
6  #include <algorithm>
7  #include <string>
8  #include <cctype>
9
10 using namespace std;
11
12 int main() {
13     int N, K;
14     cin >> N >> K;
15     cin.ignore();
16     unordered_map<string, int> Count;
17     unordered_set<char> res_set;
18
19     for (int i = 0; i < N; ++i) {
20         string words;
21         getline(cin, words);
22
23         transform(words.begin(), words.end(), words.begin(), ::tolower);
24         Count[words]++;
25     }
26     for (const auto &entry : Count) {
27         if (entry.second >= K) {
28             for (char c : entry.first) {
29                 res_set.insert(c);
30             }
31         }
32     }
33     vector<char> res(res_set.begin(), res_set.end());
34     sort(res.rbegin(), res.rend());
35     int M = res.size();
36     if (res.empty()) {
37         cout << "Empty!" << endl;
38     } else {
39         cout << M << endl;
40         for (char c : res) {
41             cout << c << " ";
42         }
43         cout << endl;
44     }
45     return 0;
46 }
47
48
```

## Результат:

```
5 2
stugna
neptune
grim
oplot
Grim
4
r m i g
PS D:\Epics>
```

Created	Compiler	Result	Time (sec.)	Memory (MiB)	Actions
7 hours ago	C++ 23	Accepted	0.035	1.297	<a href="#">View</a>

Час виконання ~ 1 год 20 хв

## Завдання 7: Practice Task 1

### Умова:

*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.

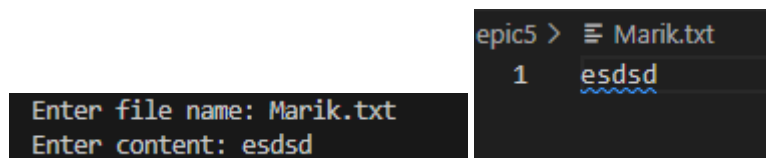
## Розв'язок:

```

1 #include <iostream>
2 #include <fstream>
3 #include <string>
4 using namespace std;
5
6 enum FileOpResult {
7     Success,
8     Failure
9 };
10
11 FileOpResult write_to_file(const char *name, const char *content) {
12     if (name == nullptr || content == nullptr || strlen(name) == 0 || strlen(content) == 0) {
13         return Failure;
14     }
15     ofstream file(name);
16     if (!file.is_open()) {
17         cerr << "Error: Could not open the file: " << name << endl;
18         return Failure;
19     }
20
21     try {
22         file.exceptions(ios::failbit | ios::badbit);
23         file << content;
24     } catch (const ios_base::failure &e) {
25         cerr << "Error: writing to file: " << e.what() << endl;
26         file.close();
27         return Failure;
28     }
29     file.close();
30     return Success;
31 }
32
33 int main() {
34     char file_name[100] = {};
35     char content[250] = {};
36
37     cout << "Enter file name: ";
38     cin.getline(file_name, 100);
39
40     if (cin.fail() || strlen(file_name) == 0) {
41         cerr << "Error: Invalid file name.\n";
42         return 1;
43     }
44
45     cout << "Enter content: ";
46     cin.getline(content, 250);
47
48     if (cin.fail() || strlen(content) == 0) {
49         cerr << "Error: Invalid content.\n";
50         return 1;
51     }
52
53     FileOpResult result = write_to_file(file_name, content);
54     if (result == Success) {
55         cout << "The file has been written successfully.\n";
56     } else {
57         cerr << "Error: Failed to write to file.\n";
58     }
59
60     return 0;
61 }

```

## Результат:



Час виконання ~ 1 год

## Завдання 8: Practice Task 2

## Умова:

*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.

## Розв'язок:

```
1  #include <iostream>
2  #include <fstream>
3  #include <cstring>
4  using namespace std;
5
6  enum FileOpResult {
7      Success,
8      Failure
9  };
10
11  FileOpResult write_to_file(const char *name, const char *content) {
12      if (name == nullptr || content == nullptr || strlen(name) == 0 || strlen(content) == 0) {
13          return Failure;
14      }
15      ofstream file(name);
16      if (!file.is_open()) {
17          cerr << "Error: Could not open the file: " << name << endl;
18          return Failure;
19      }
20
21      try {
22          file.exceptions(ios::failbit | ios::badbit);
23          file << content;
24      } catch (const ios_base::failure &e) {
25          cerr << "Error writing to file: " << e.what() << endl;
26          file.close();
27          return Failure;
28      }
29      file.close();
30      return Success;
31  }
32
33  int main() {
34      char file_name[100] = {};
35      char content[250] = {};
36
37      cout << "Enter file name: ";
38      cin.getline(file_name, 100);
39
40      if (cin.fail() || strlen(file_name) == 0) {
41          cerr << "Error: Invalid file name.\n";
42          return 1;
43      }
44
45      cout << "Enter content: ";
```

```

45     cin.getline(content, 256);
46
47     if (cin.fail() || strlen(content) == 0) {
48         cerr << "Error: Invalid content.\n";
49         return 1;
50     }
51
52     FileOpResult result = write_to_file(file_name, content);
53     if (result == Success) {
54         cout << "The file has been written successfully.\n";
55     } else {
56         cerr << "Error: Failed to write to file.\n";
57     }
58
59     return 0;
60 }
61

```

## Результат:

The first screenshot shows a command prompt window with the following text:

```

dsdskjds dsdjkjsd
dsdsds
Success!
PS D:\Epic5>

```

The second and third screenshots show the output of the program in a window titled 'practice\_work\_team\_tasks\_1\_markiiian\_lytvyn.cpp'. The output is:

```

1 123
2 Epic5 NULP.AI
3 3031 20203.5 1231

```

Час виконання ~ 45 хв

## Завдання 9: Self Practice Task Algotester Lab4v3

### Умова:

Вам дано масив, який складається з NN 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.

### Input

У першому рядку NN - кількість чисел.

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

## Output

У першому рядку ММ - кількість чисел у масиву  
У другому рядку ММ посортованих за умовою чисел.

## Розв'язок:

```
1 #include <iostream>
2 #include <vector>
3 #include <algorithm>
4 #include <set>
5 using namespace std;
6
7 int main() {
8     int N;
9     cin >> N;
10    vector<int> arr(N);
11    for (int i = 0; i < N; ++i) {
12        cin >> arr[i];
13    }
14
15    set<int> unique_set(arr.begin(), arr.end());
16    vector<int> unique_arr(unique_set.begin(), unique_set.end());
17
18    vector<int> mod0, mod1, mod2;
19    for (int num : unique_arr) {
20        if (num % 3 == 0) {
21            mod0.push_back(num);
22        } else if (num % 3 == 1) {
23            mod1.push_back(num);
24        } else {
25            mod2.push_back(num);
26        }
27    }
28
29    vector<int> M;
30
31    sort(mod0.begin(), mod0.end());
32    M.insert(M.end(), mod0.begin(), mod0.end());
33    sort(mod1.rbegin(), mod1.rend());
34    M.insert(M.end(), mod1.begin(), mod1.end());
35    sort(mod2.begin(), mod2.end());
36    M.insert(M.end(), mod2.begin(), mod2.end());
37
38    M.erase(unique(M.begin(), M.end()), M.end());
39
40    cout << M.size() << endl;
41    for (int num : M) {
42        cout << num << " ";
43    }
44    cout << endl;
45    return 0;
46 }
```

## Результат:

```
10 33 4 8 6 5 2 7 5 0
1
9
0 6 33 7 4 1 2 5 8
PS D:\Epics>
```

Created	Compiler	Result	Time (sec.)	Memory (MiB)	Actions
an hour ago	C++ 23	Accepted	0.003	1.211	<a href="#">View</a>

Час виконання ~ 35 хв

## Зустріч із командою:



## **Висновок:**

У цьому епіку я навчився працювати з файлами, символами і рядковими змінними та текстовими файлами, стандартною бібліотекою.