



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-11

Єдинець Євген Русланович

Тема роботи:

Вивчення роботи з файлами на мові C++: текстові файли, бінарні файли, символи та рядкові змінні, стандартна бібліотека для роботи з файлами, створення й використання статичних і динамічних бібліотек.

Мета роботи:

Ознайомитися з роботою з файлами в C++, зокрема з текстовими та бінарними файлами, символами та рядковими змінними. Набути навичок використання стандартної бібліотеки для роботи з файлами. Вивчити створення та використання статичних і динамічних бібліотек, а також алгоритми обробки файлів для створення ефективних і зрозумілих програм.

Теоретичні відомості:

1. Робота з файлами

<https://www.youtube.com/watch?v=SSNJ7alki-E&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=171>

2. STL - Вступ

<https://www.youtube.com/watch?v=L7JGsi4sryc&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=151>

3. STL - vector

<https://www.youtube.com/watch?v=ue-7OS3x1as&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=152>

4. Стандартна бібліотека C++

<https://www.youtube.com/watch?v=m-WJikuZGuU&list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&index=79>

Виконання роботи:

1) Опрацювання завдання та вимог до програм та середовища:

Завдання 1

VNS Lab 6

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.

21. Знищити всі парні слова у речені.

Завдання 2

VNS Lab 8

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.

21. Структура "Автомобіль":

- марка;
- серійний номер;
- реєстраційний номер;
- рік випуску.

Знищити 3 елементи з початку файлу, додати елемент поле елемента із зазначеним реєстраційним номером.

Завдання 3

VNS Lab 9

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

21.

- 1) Скопіювати з файлу F1 у файл F2 всі рядки, у яких більше 2 слів.
- 2) Визначити номер слова, у якому найбільше голосних букв.

Завдання 4

Algotester Lab 4

<https://algotester.com/uk/ContestProblem/DisplayWithEditor/135599>

Завдання 5

Algotester Lab 6

<https://algotester.com/uk/ContestProblem/DisplayWithEditor/135606>

Завдання 6

Class Practice Work

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.

Мета задачі

Розуміння методів роботи з файлами: Робота з файлами є одним з базових навиків програмування. Реалізація функції створення та запису в файл допоможе освоїти практичні навички роботи з файлами з використанням стандартної бібліотеки C++. Для виконання завдання студент має навчитись використовувати методи відкриття файла, запису масиву даних у файл, закриття файла та обробки помилок чи станів операції на кожному з етапів.

Розвиток алгоритмічне мислення: Запис у файл включає набір операцій, які якнайкраще вкладаються в концепцію алгоритма, як списку детальних кроків. Імплементация цієї функції наочно демонструє створення алгоритмів у програмуванні.

Освоїти навички роботи з текстовими стрічками: завдання допоможе освоїти роботу з C стрічка, які є масивами з нульовим символом в кінці. Типові концепції при

роботі з C стрічками це арифметика вказівників, ітерація по стрічці, копіювання частини стрічки, розбиття на токени по заданому символу.

Розвинути навички розв'язувати задачі: Запис у файл може супроводжуватись набором станів (немає доступу на створення, недостатньо місця, ін.), які необхідно передбачити у алгоритмі. Аналіз цих станів дозволяє розвинути навик розв'язання інженерних задач у програмуванні.

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.

Мета задачі

Розуміння методів роботи з файлами: Робота з файлами є одним з базових навиків програмування. Реалізація функції копіювання вмісту файла допоможе освоїти практичні навички роботи з файлами з використанням стандартної бібліотеки C++. Для виконання завдання студент має навчитись використовувати методи відкриття файла, читання вмісту файла, запису масиву даних у файл, закриття файла та обробки помилок чи станів операції на кожному з етапів.

Розвиток алгоритмічне мислення: Читання та запис у файл включає набір операцій, які якнайкраще вкладаються в концепцію алгоритма, як списку детальних кроків. Імплементация цієї функції наочно демонструє створення алгоритмів у програмуванні.

Освоїти навички роботи з потоком даних: завдання допоможе освоїти роботу з потоками даних (концепція реалізована в STL як набір класів *stream* - fstream, stringstream, streambuf та ін.). Концепція потоку даних дозволяє абстрагувати роботу з джерелами та приймачами даних та писати з її допомогою високорівневий код.

Розвинути навички розв'язувати задачі: Операції читання з файла та запис у файл можуть супроводжуватись набором різних станів (немає доступу на читання чи створення, недостатньо місця, ін.), які необхідно передбачити у алгоритмі. Аналіз цих станів дозволяє розвинути навик розв'язання інженерних задач у програмуванні.

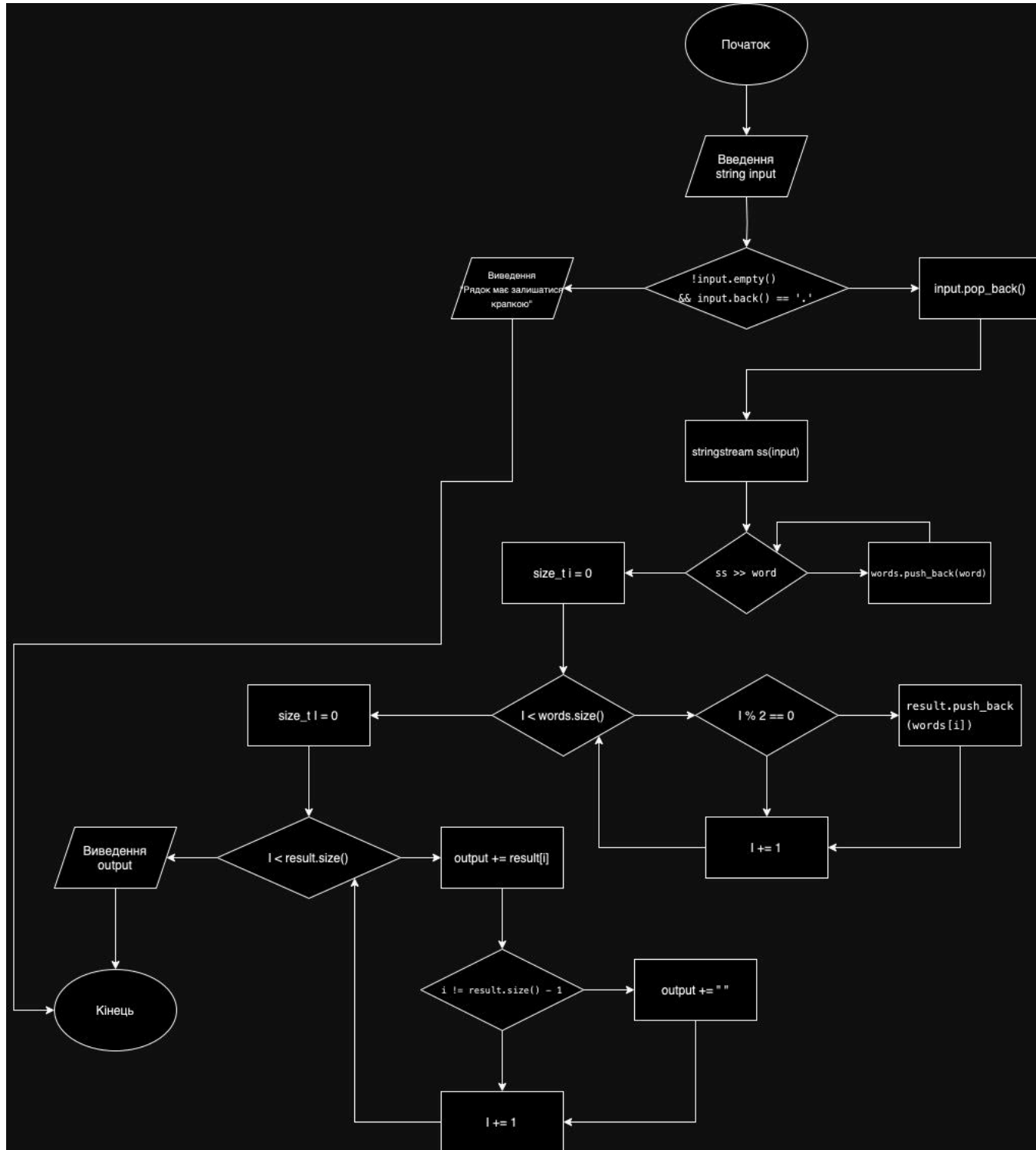
Завдання 7

Self Practice Work

<https://algotester.com/uk/ContestProblem/DisplayWithEditor/135600>

2) Дизайн та планована оцінка часу виконання завдань:

Завдання 1



Планований час виконання: 1 год. - 1 год. 30 хв.

3) Код програм:

Завдання 1:

```

1  #include <iostream>
2  #include <string>
3  #include <sstream>
4  #include <vector>
5
6  using namespace std;
7
8  int main() {
9      string input;
10     getline(cin, input);
11
12     if (!input.empty() && input.back() == '.')
13         input.pop_back();
14     else {
15         cout << "Рядок має закінчуватися крапкою." << endl;
16         return 0;
17     }
18
19     stringstream ss(input);
20     vector<string> words, result;
21     string word;
22
23     while (ss >> word)
24         words.push_back(word);
25
26     for (size_t i = 0; i < words.size(); ++i)
27         if (i % 2 == 0)
28             result.push_back(words[i]);
29
30     string output;
31     for (size_t i = 0; i < result.size(); ++i) {
32         output += result[i];
33         if (i != result.size() - 1)
34             output += " ";
35     }
36
37     cout << output << "." << endl;
38
39     return 0;
40 }

```

Завдання 2:

```

1  #include <iostream>
2  #include <fstream>
3  #include <vector>
4
5  using namespace std;
6
7  struct Car {
8      char brand[50];
9      int serial_number;
10     char reg_number[20];
11     int year;
12 };
13
14 void addCarToFile(const string& filename, const Car& car) {
15     ofstream file(filename, ios::binary | ios::app);
16     if (file) file.write(reinterpret_cast<const char*>(&car), sizeof(Car));
17 }
18
19 void printCarsFromFile(const string& filename) {
20     ifstream file(filename, ios::binary);
21     Car car;
22     while (file.read(reinterpret_cast<char*>(&car), sizeof(Car))) {
23         cout << car.brand << " " << car.serial_number << " " << car.reg_number << " " << car.year << endl;
24     }
25 }
26
27 void deleteFirstThreeCars(const string& filename) {
28     ifstream file(filename, ios::binary);
29     vector<Car> cars;
30     Car car;
31     while (file.read(reinterpret_cast<char*>(&car), sizeof(Car))) cars.push_back(car);
32     file.close();
33     ofstream outFile(filename, ios::binary | ios::trunc);
34     for (size_t i = 3; i < cars.size(); ++i) outFile.write(reinterpret_cast<const char*>(&cars[i]), sizeof(Car));
35 }
36
37 void addCarAfterRegNumber(const string& filename, const string& targetRegNumber, const Car& newCar) {
38     ifstream file(filename, ios::binary);
39     vector<Car> cars;
40     Car car;
41     while (file.read(reinterpret_cast<char*>(&car), sizeof(Car))) cars.push_back(car);
42     file.close();
43     ofstream outFile(filename, ios::binary | ios::trunc);
44     for (const auto& c : cars) {
45         outFile.write(reinterpret_cast<const char*>(&c), sizeof(Car));
46         if (targetRegNumber == c.reg_number) outFile.write(reinterpret_cast<const char*>(&newCar), sizeof(Car));
47     }
48 }
49
50 Car createCar() {
51     Car car;
52     cin.ignore();
53     cin.getline(car.brand, 50);
54     cin >> car.serial_number;
55     cin.ignore();
56     cin.getline(car.reg_number, 20);
57     cin >> car.year;
58     return car;
59 }
60
61 int main() {
62     const string filename = "cars.dat";
63     int n;
64     cin >> n;
65     for (int i = 0; i < n; ++i) addCarToFile(filename, createCar());
66     printCarsFromFile(filename);
67     deleteFirstThreeCars(filename);
68     printCarsFromFile(filename);
69     string regNum;
70     cin >> regNum;
71     addCarAfterRegNumber(filename, regNum, createCar());
72     printCarsFromFile(filename);
73
74     return 0;
75 }

```

Завдання 3:


```

1  #include <iostream>
2  #include <fstream>
3  #include <string>
4  #include <sstream>
5  #include <cctype>
6
7  using namespace std;
8
9  int countVowels(const string& word) {
10     int count = 0;
11     string vowels = "aeiouAEIOU";
12     for (char c : word) {
13         if (vowels.find(c) != string::npos) {
14             count++;
15         }
16     }
17     return count;
18 }
19
20 int main() {
21     ofstream fileF1("F1.txt");
22     fileF1 << "This is a sample line." << endl;
23     fileF1 << "Hello world!" << endl;
24     fileF1 << "C++ programming is fun." << endl;
25     fileF1 << "Short sentence." << endl;
26     fileF1 << "Another example of text." << endl;
27     fileF1 << "How are you today?" << endl;
28     fileF1 << "A simple test case." << endl;
29     fileF1 << "One more line to go." << endl;
30     fileF1 << "We need more lines." << endl;
31     fileF1 << "Finally, the last line." << endl;
32     fileF1.close();
33
34     ifstream fileIn("F1.txt");
35     ofstream fileF2("F2.txt");
36
37     string line;
38     vector<string> words;
39     int maxVowelCount = 0;
40     int wordIndexWithMostVowels = -1;
41
42     while (getline(fileIn, line)) {
43         istringstream iss(line);
44         words = vector<string>(istream_iterator<string>(iss), {});
45         if (words.size() > 2) {
46             fileF2 << line << endl;
47         }
48
49         for (size_t i = 0; i < words.size(); ++i) {
50             int vowelCount = countVowels(words[i]);
51             if (vowelCount > maxVowelCount) {
52                 maxVowelCount = vowelCount;
53                 wordIndexWithMostVowels = i + 1;
54             }
55         }
56     }
57
58     fileIn.close();
59     fileF2.close();
60
61     cout << "Номер слова з найбільшою кількістю голосних: " << wordIndexWithMostVowels << endl;
62
63     return 0;
64 }

```

Завдання 4:

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

```

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

```

Завдання 5

```

1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  int main() {
7      int gridSize, queryCount;
8      cin >> gridSize;
9
10     int grid[gridSize][gridSize];
11     int possibleNumbers[gridSize + 1];
12
13     for (int i = 1; i <= gridSize; i++) {
14         possibleNumbers[i] = i;
15     }
16
17     for (int row = 0; row < gridSize; row++) {
18         string inputRow;
19         cin >> inputRow;
20         for (int col = 0; col < gridSize; col++) {
21             grid[row][col] = inputRow[col] - '0';
22         }
23     }
24
25     cin >> queryCount;
26     int queryCoordinates[queryCount][2];
27
28     for (int i = 0; i < queryCount; i++) {
29         cin >> queryCoordinates[i][0] >> queryCoordinates[i][1];
30         queryCoordinates[i][0]--;
31         queryCoordinates[i][1]--;
32     }
33
34     for (int i = 0; i < queryCount; i++) {
35         int queryRow = queryCoordinates[i][0];
36         int queryCol = queryCoordinates[i][1];
37
38         if (grid[queryRow][queryCol] == 0) {
39             int availableNumbers[gridSize + 1];
40             for (int j = 1; j <= gridSize; j++) {
41                 availableNumbers[j] = possibleNumbers[j];
42             }
43
44             for (int j = 0; j < gridSize; j++) {
45                 if (grid[queryRow][j] > 0) {
46                     availableNumbers[grid[queryRow][j]] = 0;
47                 }
48                 if (grid[j][queryCol] > 0) {
49                     availableNumbers[grid[j][queryCol]] = 0;
50                 }
51             }
52
53             int count = 0;
54             for (int j = 1; j <= gridSize; j++) {
55                 if (availableNumbers[j] != 0) {
56                     count++;
57                 }
58             }
59
60             cout << count << endl;
61             for (int j = 1; j <= gridSize; j++) {
62                 if (availableNumbers[j] != 0) {
63                     cout << availableNumbers[j] << " ";
64                 }
65             }
66             cout << endl << endl;
67         } else {
68             cout << "1" << endl;
69             cout << grid[queryRow][queryCol] << endl << endl;
70         }
71     }
72
73     return 0;
74 }

```

Завдання 6

```
1  #include <iostream>
2  #include <fstream>
3  #include <cstring>
4
5  using namespace std;
6
7  enum FileOpResult { Success, Failure };
8
9  FileOpResult write_to_file(const char* name, const char* content) {
10     if (!name || !content) {
11         cerr << "Error: Invalid input parameters." << endl;
12         return Failure;
13     }
14
15     ofstream file(name);
16     if (!file.is_open()) {
17         cerr << "Error: Failed to open the file." << endl;
18         return Failure;
19     }
20
21     file << content;
22     if (file.fail()) {
23         cerr << "Error: Failed to write to the file." << endl;
24         file.close();
25         return Failure;
26     }
27
28     file.close();
29     if (file.fail()) {
30         cerr << "Error: Failed to close the file." << endl;
31         return Failure;
32     }
33
34     return Success;
35 }
36
37 int main() {
38     char filename[256];
39     char content[1024];
40
41     cout << "Enter the filename: ";
42     cin.getline(filename, 256);
43
44     cout << "Enter the content to write into the file: ";
45     cin.getline(content, 1024);
46
47     FileOpResult result = write_to_file(filename, content);
48     if (result == Success) {
49         cout << "File written successfully." << endl;
50     } else {
51         cout << "File operation failed." << endl;
52     }
53
54     return 0;
55 }
```

Завдання 7

```

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

```

```

1  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5
6  void insertionSort(vector<int>& arr, bool ascending) {
7      for (size_t i = 1; i < arr.size(); i++) {
8          int key = arr[i];
9          int j = i - 1;
10         while (j >= 0 && ((ascending && arr[j] > key) || (!ascending && arr[j] < key))) {
11             arr[j + 1] = arr[j];
12             j--;
13         }
14         arr[j + 1] = key;
15     }
16 }
17
18 void removeDuplicates(vector<int>& arr) {
19     vector<int> uniqueArr;
20     for (int x : arr) {
21         if (uniqueArr.empty() || uniqueArr.back() != x) {
22             uniqueArr.push_back(x);
23         }
24     }
25     arr = uniqueArr;
26 }
27
28 int main() {
29     int N;
30     cin >> N;
31     vector<int> arr(N);
32
33     for (int i = 0; i < N; i++) {
34         cin >> arr[i];
35     }
36
37     vector<int> mod0, mod1, mod2;
38     for (int x : arr) {
39         if (x % 3 == 0)
40             mod0.push_back(x);
41         else if (x % 3 == 1)
42             mod1.push_back(x);
43         else
44             mod2.push_back(x);
45     }
46
47     insertionSort(mod0, true);
48     insertionSort(mod1, false);
49     insertionSort(mod2, true);
50
51     vector<int> result;
52     result.insert(result.end(), mod0.begin(), mod0.end());
53     result.insert(result.end(), mod1.begin(), mod1.end());
54     result.insert(result.end(), mod2.begin(), mod2.end());
55
56     removeDuplicates(result);
57
58     cout << result.size() << endl;
59     for (int x : result) {
60         cout << x << " ";
61     }
62     cout << endl;
63
64     return 0;
65 }

```

4) Результат виконання завдань, тестування та фактично затрачений час:

Завдання 1:

Рядок має закінчуватися крапкою.
Рядок закінчуватися.

Фактично затрачений час: 1 год 10 хв.

Завдання 2:

```

Enter number of cars to add: 2
Enter car details (brand, serial_number, reg_number, year):
BMW
1234
AB12
2000
Added car to file: BMW 1234 AB12 2000
Enter car details (brand, serial_number, reg_number, year):
Ford
4321
CD34
1998
Added car to file: Ford 4321 CD34 1998
Current cars in file:
BMW 1234 AB12 2000
Ford 4321 CD34 1998

Deleting the first three cars from the file...
Deleted all cars from the file since there are 3 or fewer cars.
Current cars in file:

Enter registration number after which to add a new car: CD34
Enter car details (brand, serial_number, reg_number, year):
Toyota
1324
QW12
1990
Registration number CD34 not found. New car not added.
Current cars in file:

```

Фактично затрачений час: 2 год.

Завдання 3:

```

Word with the most vowels: programming (vowels: 3)

```

Фактично затрачений час: 1 год. 15 хв.

Завдання 4:

C++ 23	Зараховано	0.003	1.328
--------	------------	-------	-------

<https://algotester.com/uk/ContestProblem/DisplayWithEditor/135599>

Фактично затрачений час: 1 год. 30 хв.

Завдання 5:

C++ 23	Зараховано	0.003	1.379
--------	------------	-------	-------

<https://algotester.com/uk/ContestProblem/DisplayWithEditor/135606>

Фактично затрачений час: 1 год. 45 хв.

Завдання 6:


```
Enter the filename: File_Name
Enter the content to write into the file: Hello, World!
File written successfully.
```

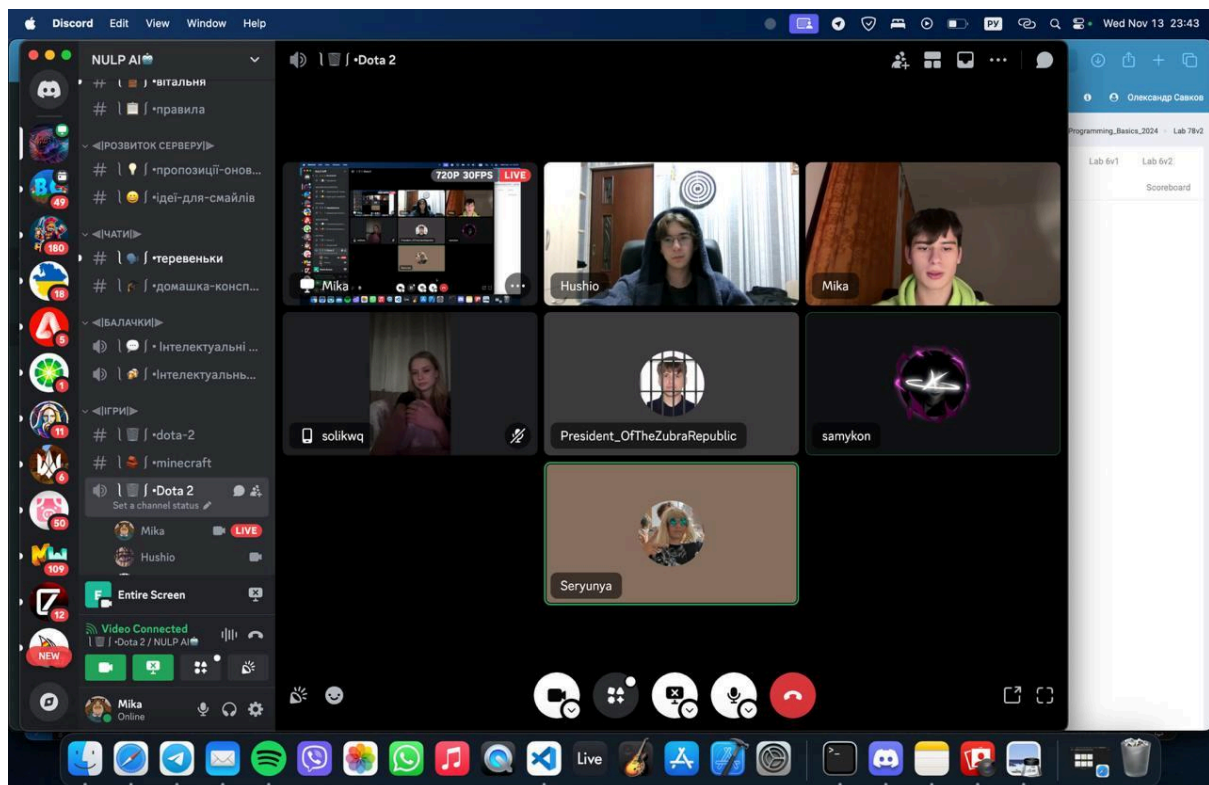
Фактично затрачений час: 60 хв.

Завдання 7:

C++ 23	Зараховано	0.003	1.203
--------	------------	-------	-------

<https://algotester.com/uk/ContestProblem/DisplayWithEditor/135600>

Робота з командою:



Висновок:

Під час виконання цього епіку я вивчив роботу з файлами в C++, зокрема текстові та бінарні файли, а також їх використання для зберігання та обробки даних. Я засвоїв, як працювати із символами та рядковими змінними, а також ефективно використовувати стандартну бібліотеку для роботи з файлами. Окрім цього, я ознайомився зі створенням та використанням статичних і динамічних бібліотек, а також алгоритмами обробки файлів, що дозволило створювати більш складні та гнучкі програми.