

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-11

Кравченко Артем Миколайович

**Тема:**

Файли. Бінарні Файли. Символи і Рядкові Змінні та Текстові Файли. Стандартна бібліотека та деталі/методи роботи з файлами. Створення й використання бібліотек.

**Мета:**

Ознайомитися з основами роботи з файлами, зокрема з бінарними і текстовими файлами, символами та рядковими змінними, а також вивчити основні методи та функції стандартної бібліотеки для маніпуляцій з файлами. Студенти навчатися створювати та використовувати власні бібліотеки для оптимізації коду й закріплять навички роботи з файлами через практичні завдання.

**Теоретичні відомості:**

1. Вступ до Роботи з Файлами
2. Символи і Рядкові Змінні
3. Текстові Файли
4. Бінарні Файли
5. Стандартна бібліотека та робота з файлами
6. Створення й використання бібліотек

**Індивідуальний план опрацювання теорії:**

1. <https://acode.com.ua/urok-220-bazovyj-fajlovyj-vvid-i-vyvid/>
2. <https://www.youtube.com/watch?v=CUOJCNH3hS4>
3. <https://www.youtube.com/watch?v=FeNqHytI0fA>
4. <https://www.youtube.com/watch?v=FeNqHytI0fA>
5. <https://www.youtube.com/watch?v=m-WJikuZGuU>
6. [https://plug-blog.blogspot.com/2016/05/c\\_31.html](https://plug-blog.blogspot.com/2016/05/c_31.html)

## **Виконання роботи:**

### **Завдання 1: VNS Lab 6 - Task 1. Варіант - 24**

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.

24. Для рядка знайти кількість слів у ньому.

### **Завдання 2: VNS Lab 8 - Task 1. Варіант - 24**

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.

Знищити всі елементи, у яких рік будівлі менше заданого, додати 2 елементи перед елементом із зазначеним номером.

24. Структура "Студент":

- прізвище, ім'я, по батькові;
- номер телефону;
- група;
- оцінки по 3 основних предметах.

Знищити всі елементи із групи із зазначеним номером, у яких середнє арифметичне оцінок менше заданого, додати елемент після елемента із заданим прізвищем.

### **Завдання 3: VNS Lab 9 - Task 1. Варіант - 24**

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

24.

1) Скопіювати з файлу F1 у файл F2 всі рядки, у яких міститься не менше двох однакових слів.

2) Визначити номер слова, у якому найбільше цифр.

### **Завдання 4: Algotester lab 4 Варіант 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  $[4, 5, 6, 7, 1, 2, 3]$ . **(З використанням засобів STL)**

### **Завдання 5: Algotester lab 4 Варіант 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  $[4, 5, 6, 7, 1, 2, 3]$ . **(Без використанням засобів STL)**

### **Завдання 6: Algotester lab 6 Варіант 2**

У вас є шахова дошка розміром  $8 \times 8$  та дуже багато фігур. Кожна клітинка може мати таке значення:

- Пуста клітинка O
- Пішак P
- Тура R
- Кінь N
- Слон B
- Король K
- Королева Q

Вам дають позиції фігур на дошці (всі фігури одного кольору, кількість королів може бути  $> 1$ ). Далі йдуть  $Q$  запитів з координатами клітинки  $\{x, y\}$ . На кожен запит ви маєте вивести стрічку  $si$  - посортовані за алфавітом букви фігур, які атакують цю клітинку (пішаки атакують вниз). У випадку, якщо на клітинці стоїть якась фігура - виведіть символ X. У

випадку, якщо клітинку не атакують - виведіть О. Наявність фігури у певній клітинці не блокує атаку для іншої фігури. Тобто якщо між турою та клітинкою стоїть інша фігура - вважається що тура атакує цю клітинку.

## Завдання 7: Class Practice Work

*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.

*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 8: Self Practice Work

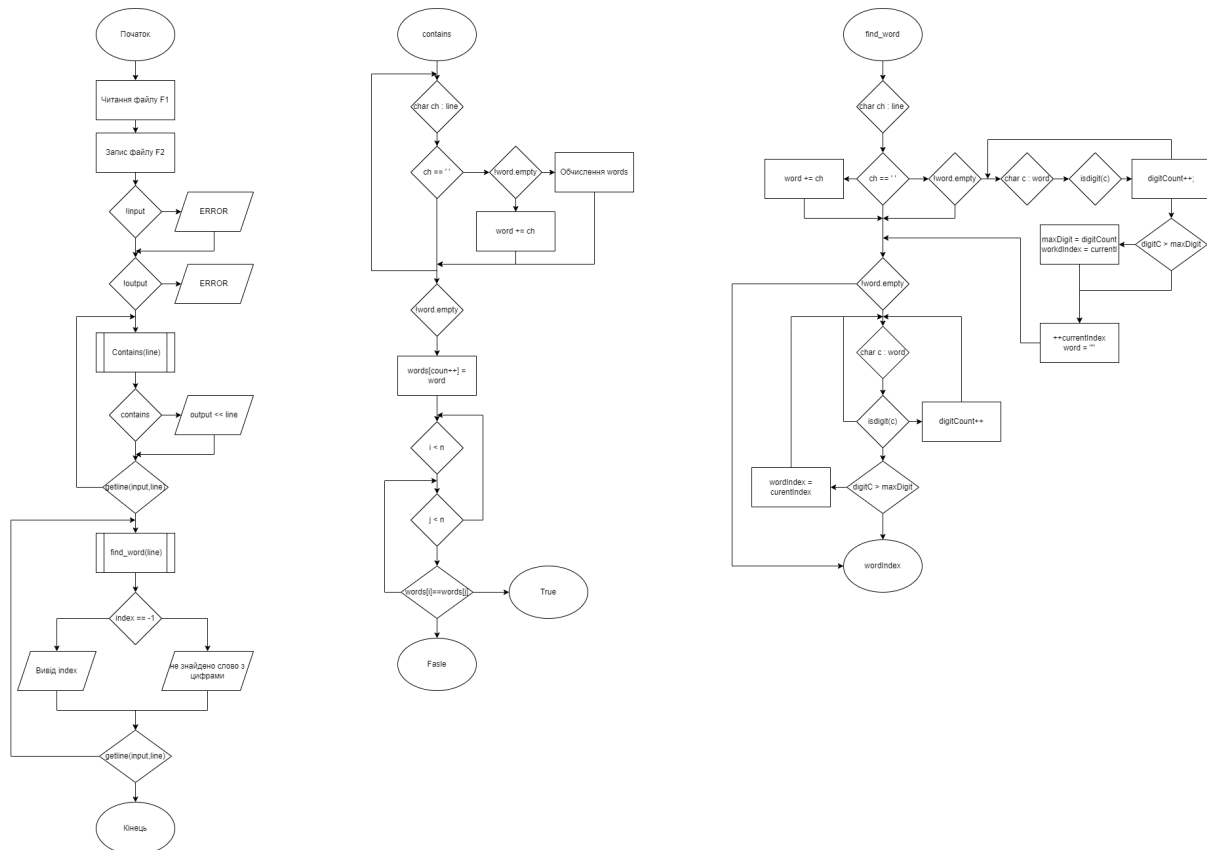
До Тойлет-мена, відомого вам білоруського супергероя, дуже часто звертаються по допомогу різні люди. Проте, очевидно, усім мужній гігант допомогти не зможе — не вистачить часу. Та й не дуже хоче. Саме тому, коли до нього приходить певне SMS-повідомлення з проханням про допомогу, він погодиться допомогти тоді й лише тоді, коли в цьому повідомленні знайдеться хоча б k входжень рядка **TOILET**, які не перетинаються. За заданим повідомленням s, яке складається з великих латинських символів, виведіть **YES**, якщо Тойлет-мен погодиться допомогти людині, яка написала це повідомлення. У протилежному разі виведіть **NO**.

**Дизайн та планувальна оцінка часу виконання завдань:**

**Завдання 1:** Запланований час виконання 1 година.

**Завдання 2:** Запланований час виконання 1-1.5 години.

**Завдання 3:**



Запланований час виконання 40-50 хвилин.

**Завдання 4:** Запланований час виконання 1 година.

**Завдання 5:** Запланований час виконання 1-2 години.

**Завдання 6:** Запланований час виконання 1-2 години.

**Завдання 7:** Запланований час виконання 1-1.5 години.

**Завдання 8:** Запланований час виконання 30-40 хвилин.

## Код програм з посиланням на зовнішні ресурси:

### Завдання 1:

```
1  #include <iostream>
2  #include <stdio.h>
3
4  using namespace std;
5
6  int main(){
7      const int max_lenth = 255;
8      char s[max_lenth + 1];
9      cout << "Введіть слова: ";
10     gets(s);
11
12     int lenth = 0;
13
14     while (s[lenth] != '\0')
15     {
16         lenth++;
17     }
18
19     int count = 0;
20     bool inWord = false;
21
22     for (int i = 0; i < lenth; i++)
23     {
24         if (s[i] != ' ' || s[i] != '.' || s[i] != '\t')
25         {
26             if (!inWord)
27             {
28                 inWord = true;
29                 count++;
30             }else if(s[i] == ' ' || s[i] == '.' || s[i] == '\t'){
31                 inWord = false;
32             }
33         }
34     }
35
36     cout << "Кількість слів  рядку: " << count;
37
38     return 0;
```

[https://github.com/artificial-intelligence-department/ai\\_programming\\_playground\\_2024/pull/332/files#diff-8bc63c953797fc6ee364029dc614a2a7f529ac0f7b13bd4d0ee8b0f63195efe1](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/332/files#diff-8bc63c953797fc6ee364029dc614a2a7f529ac0f7b13bd4d0ee8b0f63195efe1)

## Завдання 2:

```
1  #include <iostream>
2  #include <fstream>
3  #include <vector>
4  #include <string>
5
6  using namespace std;
7
8  struct Stud {
9      string name;
10     string phone;
11     string grp;
12     int marks[3];
13 };
14
15 void addStud(vector<Stud>& studs) {
16     Stud stud;
17     cout << "Введіть повне ім'я: ";
18     getline(cin, stud.name);
19     cout << "Введіть номер телефону: ";
20     getline(cin, stud.phone);
21     cout << "Введіть групу: ";
22     getline(cin, stud.grp);
23     cout << "Введіть оцінки для трьох предметів через пробіл: ";
24     cin >> stud.marks[0] >> stud.marks[1] >> stud.marks[2];
25     cin.ignore();
26     studs.push_back(stud);
27 }
28
29 void deleteStud(vector<Stud>& studs, const string& grpNum, float minAvg) {
30     for (auto it = studs.begin(); it != studs.end(); ) {
31         float average = (it->marks[0] + it->marks[1] + it->marks[2]) / 3.0;
32         if (it->grp == grpNum && average < minAvg) {
33             it = studs.erase(it);
34         } else {
35             ++it;
36         }
37     }
38 }
39
```



```

39
40 void saveToFile(const vector<Stud>& studs, const string& fileName) {
41     ofstream file(fileName, ios::binary);
42     if (!file) {
43         cerr << "Помилка відкриття файлу для запису." << endl;
44         return;
45     }
46     for (const auto& stud : studs) {
47         file << stud.name << '\n' << stud.phone << '\n' << stud.grp << '\n';
48         for (int mark : stud.marks) {
49             file << mark << ' ';
50         }
51         file << '\n';
52     }
53 }
54
55 void loadFromFile(vector<Stud>& studs, const string& fileName) {
56     ifstream file(fileName, ios::binary);
57     if (!file) {
58         cerr << "Помилка відкриття файлу для читання." << endl;
59         return;
60     }
61     Stud stud;
62     while (getline(file, stud.name)) {
63         getline(file, stud.phone);
64         getline(file, stud.grp);
65         for (int& mark : stud.marks) {
66             file >> mark;
67         }
68         file.ignore();
69         studs.push_back(stud);
70     }
71 }
72
73 void printStuds(const vector<Stud>& studs) {
74     for (const auto& stud : studs) {
75         cout << "Повне ім'я: " << stud.name << "\n";
76         cout << "Номер телефону: " << stud.phone << "\n";

```

```

77     cout << "Група: " << stud.grp << "\n";
78     cout << "Оцінки: " << stud.marks[0] << ", " << stud.marks[1] << ", " << stud.marks[2] << "\n";
79 }
80 }
81
82 int main() {
83     vector<Stud> studs;
84     string fileName = "students.txt";
85     int choice;
86
87     while (true) {
88         cout << "1. Додати студента\n2. Видалити студента\n3. Вивести студентів\n4. Зберегти файл\n5. Завантажити файл\n";
89         cin >> choice;
90         cin.ignore();
91         switch (choice) {
92             case 1:
93                 addStud(studs);
94                 break;
95             case 2: {
96                 string grpNum;
97                 float minAvg;
98                 cout << "Введіть номер групи: ";
99                 getline(cin, grpNum);
100                 cout << "Введіть мінімальний середній бал: ";
101                 cin >> minAvg;
102                 cin.ignore();
103                 deleteStud(studs, grpNum, minAvg);
104                 break;
105             }
106             case 3:
107                 printStuds(studs);
108                 break;
109             case 4:
110                 saveToFile(studs, fileName);
111                 break;
112             case 5:

```

```

112             case 5:
113                 loadFromFile(studs, fileName);
114                 break;
115             case 6:
116                 return 0;
117             default:
118                 cout << "Невірний вибір." << endl;
119                 break;
120         }
121     }
122
123     return 0;
124 }

```

[https://github.com/artificial-intelligence-department/ai\\_programming\\_playground\\_2024/pull/332/files#diff-b84a2956a92b17e3136d7c2a5bec58e62b122dee70f57b82a32a534e34f87285](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/332/files#diff-b84a2956a92b17e3136d7c2a5bec58e62b122dee70f57b82a32a534e34f87285)

### Завдання 3:

```
1  #include <iostream>
2  #include <fstream>
3  #include <string>
4
5  using namespace std;
6
7  bool contains(const string& line) {
8      string words[100];
9      int count = 0;
10     string word = "";
11
12     for (char ch : line)
13     {
14         if (ch == ' ')
15         {
16             if (!word.empty())
17             {
18                 words[count++] = word;
19                 word = "";
20             }
21         } else {
22             word += ch;
23         }
24     }
25     if (!word.empty())
26     {
27         words[count++] = word;
28     }
29
30     for (int i = 0; i < count; i++)
31     {
32         for (int j = i + 1; j < count; j++)
33         {
34             if (words[i] == words[j])
35             {
36                 return true;
37             }
38         }
39     }
```

```
40     return false;
41 }
42
43 int find_word(const string& line) {
44     string word = "";
45     int maxDigits = 0;
46     int wordIndex = -1;
47     int currentIndex = 0;
48
49     for (char ch : line)
50     {
51         if (ch == ' ')
52         {
53             if (!word.empty()) {
54                 int digitCount = 0;
55                 for (char c : word)
56                 {
57                     if (isdigit(c))
58                     {
59                         digitCount++;
60                     }
61                 }
62                 if (digitCount > maxDigits)
63                 {
64                     maxDigits = digitCount;
65                     wordIndex = currentIndex;
66                 }
67                 ++currentIndex;
68                 word = "";
69             }
70         } else {
71             word += ch;
72         }
73     }
74
75     if (!word.empty())
76     {
```

```

77         int digitCount = 0;
78         for (char c : word)
79         {
80             if (isdigit(c))
81             {
82                 digitCount++;
83             }
84         }
85         if (digitCount > maxDigits)
86         {
87             wordIndex = currentIndex;
88         }
89     }
90
91     return wordIndex;
92 }
93
94 int main() {
95     ifstream input("F1.txt");
96     ofstream output("F2.txt");
97
98     if (!input) {
99         cerr << "Помилка: не вдалося відкрити вхідний файл." << endl;
100         return 1;
101     }
102     if (!output) {
103         cerr << "Помилка: не вдалося відкрити вихідний файл." << endl;
104         return 1;
105     }
106
107     string line;
108     while (getline(input, line))
109     {
110         if (contains(line))
111         {
112             output << line << endl;
113         }
114     }

```

```

110     if (contains(line))
111     {
112         output << line << endl;
113     }
114 }
115
116 input.clear();
117 input.seekg(0, ios::beg);
118
119 cout << "Індекс слова з найбільшою кількістю цифр у кожному рядку:" << endl;
120 while (getline(input, line))
121 {
122     int index = find_word(line);
123     if (index == -1) {
124         cout << "У рядку не знайдено слово з цифрами" << endl;
125     } else {
126         cout << index << endl;
127     }
128 }
129
130 return 0;
131 }
132

```

[https://github.com/artificial-intelligence-department/ai\\_programming\\_playground\\_2024/pull/332/files#diff-f97a4408d5d8bac9d0a4026b8e5b1aa5070a6681a4d95ed7025ae833be7a777f](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/332/files#diff-f97a4408d5d8bac9d0a4026b8e5b1aa5070a6681a4d95ed7025ae833be7a777f)

## Завдання 4:

```
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;
11     for (int i = 0; i < N; i++)
12     {
13         int x;
14         cin >> x;
15         if (find(a.begin(), a.end(), x) == a.end())
16         {
17             a.push_back(x);
18         }
19     }
20     sort(a.begin(), a.end());
21
22     K %= a.size();
23     rotate(a.begin(), a.begin() + K, a.end());
24
25     cout << a.size() << endl;
26     for (int i : a)
27     {
28         cout << i << " ";
29     }
30
31
32
33     return 0;
34 }
```

[https://github.com/artificial-intelligence-department/ai\\_programming\\_playground\\_2024/pull/332/files#diff-56d2ae5185ad5cd55265e2c1b189643a591bfa2605a9453aaf997f4340cc9193](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/332/files#diff-56d2ae5185ad5cd55265e2c1b189643a591bfa2605a9453aaf997f4340cc9193)

## Завдання 5:

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main(){
6      int N, K;
7      cin >> N >> K;
8      int a[1000];
9      int b[1000];
10     int size = 0;
11
12     for (int i = 0; i < N; i++)
13     {
14         int x;
15         cin >> x;
16         bool unique = true;
17         for (int j = 0; j < size; j++)
18         {
19             if (a[j] == x)
20             {
21                 unique = false;
22                 break;
23             }
24         }
25         if (unique)
26         {
27             a[size++] = x;
28         }
29     }
30
31
32     for (int i = 0; i < size - 1; i++)
33     {
34         for (int j = 0; j < size - 1; j++)
35         {
36             if (a[j] > a[j+1])
37             {
38                 int temp = a[j];
39                 a[j] = a[j+1];
```



```
39         a[j] = a[j+1];
40         a[j+1] = temp;
41     }
42
43 }
44
45 }
46 K %= size;
47
48 for (int i = 0; i < size; i++)
49 {
50     b[i] = a[(i + K) % size];
51 }
52
53 cout << size << endl;
54 for (int i = 0; i < size; i++)
55 {
56     cout << b[i] << " ";
57 }
58 }
59
60 return 0;
61 }
```

[https://github.com/artificial-intelligence-department/ai\\_programming\\_playground\\_2024/pull/332/files#diff-5a4153f87b79c74c40a5dbe4fdc85a44f785b4b5033eef9f7c9c72075161d2ad](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/332/files#diff-5a4153f87b79c74c40a5dbe4fdc85a44f785b4b5033eef9f7c9c72075161d2ad)

## Завдання 6:

```
1  #include <iostream>
2  #include <string>
3  #include <vector>
4  #include <algorithm>
5
6  using namespace std;
7  const int N = 8;
8  bool attacks(char piece, int pieceX, int pieceY, int x, int y){
9      switch (piece)
10     {
11         case 'R':
12             return pieceX == x || pieceY == y;
13         case 'B':
14             return abs(pieceX - x) == abs(pieceY - y);
15         case 'N':
16             return (abs(pieceX - x) == 2 && abs(pieceY - y) == 1) || (abs(pieceX - x) == 1 && abs(pieceY - y) == 2);
17         case 'P':
18             return pieceX + 1 == x && abs(pieceY - y) == 1;
19         case 'K':
20             return abs(pieceX - x) <= 1 && abs(pieceY - y) <= 1;
21         case 'Q':
22             return (pieceX == x || pieceY == y) || (abs(pieceX - x) == abs(pieceY - y));
23         default: return false;
24     }
25 }
26 int main(){
27     vector<string> board(N);
28     for (int i = 0; i < N; i++)
29     {
30         cin >> board[i];
31     }
32
33     int Q;
34     cin >> Q;
35     vector<string> result(Q);
36     for (int k = 0; k < Q; k++)
37     {
38         int x, y;
39         cin >> x >> y;
```

```

40     x--;
41     y--;
42     if (board[x][y] != 'O')
43     {
44         result[k] = "X";
45         continue;
46     }
47     string results;
48     for (int i = 0; i < N; i++)
49     {
50         for (int j = 0; j < N; j++)
51         {
52             char piece = board[i][j];
53             if (piece != 'O' && attacks(piece, i, j, x, y))
54             {
55                 bool found = false;
56                 for (char c : results)
57                 {
58                     if (c == piece)
59                     {
60                         found = true;
61                         break;
62                     }
63                 }
64                 if (!found)
65                 {
66                     results += piece;
67                 }
68             }
69         }
70     }
71 }
72
73 if (results.empty())
74 {
75     result[k] = "O";
76 } else {
77     sort(results.begin(), results.end());
78     result[k] = results;
79 }
80
81 for (const string& results : result)
82 {
83     cout << results << endl;
84 }
85
86 return 0;
87 }
88

```

[https://github.com/artificial-intelligence-department/ai\\_programming\\_playground\\_2024/pull/332/files#diff-232aba480a346f1578ebd6c932d78bbfb547d97127a73c8944aff5aadaf56b43](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/332/files#diff-232aba480a346f1578ebd6c932d78bbfb547d97127a73c8944aff5aadaf56b43)

## Завдання 7:

```
1  #include <iostream>
2  #include <string>
3  #include <fstream>
4
5  using namespace std;
6
7  enum FileOpResult { Success, Failure };
8
9  FileOpResult write(const char *name, const char *content) {
10     ofstream fout(name);
11     if (!fout.is_open())
12     {
13         return Failure;
14     }
15     fout << content;
16     if (!fout)
17     {
18         return Failure;
19     }
20     fout.close();
21     return Success;
22 }
23
24 int main() {
25     string name1 = "file_from.txt";
26     string content;
27
28     cout << "Введіть інформацію, яка буде записана у файл:" << endl;
29     getline(cin, content);
30
31     FileOpResult result = write(name1.c_str(), content.c_str());
32     if (result == Failure)
33     {
34         cout << "Failure" << endl;
35         return 1;
36     }
37     cout << "Success" << endl;
38
39     return 0;
```

[https://github.com/artificial-intelligence-department/ai\\_programming\\_playground\\_2024/pull/332/files#diff-d4759e4adbbfdc2a6ed4aef2df9f912dcb1e38167a2597d99e039c53da906c0a](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/332/files#diff-d4759e4adbbfdc2a6ed4aef2df9f912dcb1e38167a2597d99e039c53da906c0a)

```

1  #include <iostream>
2  #include <string>
3  #include <fstream>
4
5  using namespace std;
6
7  enum FileOpResult { Success, Failure };
8
9  FileOpResult copy(const char *file_from, const char *file_to) {
10     ifstream fin(file_from);
11     if (!fin.is_open())
12     {
13         return Failure;
14     }
15     ofstream fout(file_to);
16     if (!fout.is_open())
17     {
18         return Failure;
19     }
20     string line;
21     while (getline(fin, line))
22     {
23         fout << line << endl;
24     }
25     if (!fout)
26     {
27         return Failure;
28     }
29     fin.close();
30     fout.close();
31     return Success;
32 }
33
34 int main() {
35     string name1 = "file_from.txt";
36     string name2 = "file_to.txt";
37
38     FileOpResult result = copy(name1.c_str(), name2.c_str());
39     if (result == Failure)

```

```

40     {
41         cout << "Failure" << endl;
42         return 1;
43     }
44     cout << "Success" << endl;
45
46     return 0;
47 }

```

[https://github.com/artificial-intelligence-department/ai\\_programming\\_playground\\_2024/pull/332/files#diff-4252b15135b783feea531cc5cb5f63471c178dc0da4c3e35a86b2a95834820cc](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/332/files#diff-4252b15135b783feea531cc5cb5f63471c178dc0da4c3e35a86b2a95834820cc)

## Завдання 8:

```
1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  int main(){
7      int k;
8      string s;
9
10     cin >> k;
11     cin >> s;
12
13     string ryadok = "TOILET";
14     int count = 0;
15     size_t pos = 0;
16
17     while ((pos = s.find(ryadok, pos)) != -1)
18     {
19         count++;
20         pos += ryadok.length();
21     }
22     if (count >= k)
23     {
24         cout << "YES" << endl;
25     }else{
26         cout << "NO" << endl;
27     }
28     return 0;
29 }
```

[https://github.com/artificial-intelligence-department/ai\\_programming\\_playground\\_2024/pull/332/files#diff-f8437e19a0baf2c0a75d551a5a839fe01becb37cd14580c00b0e371be10a7659](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/332/files#diff-f8437e19a0baf2c0a75d551a5a839fe01becb37cd14580c00b0e371be10a7659)

## Результат виконання завдань, тестування та фактично витрачений час:

### Завдання 1:

```
Введіть слова: Hello world
Кількість слів у рядку: 2
PS D:\VS Code project\C ++> █
```

Фактично витрачений час: 1 година.

### Завдання 2:

```
1. Додати студента
2. Видалити студента
3. Вивести студентів
4. Зберегти у файл
5. Завантажити з файлу
6. Вийти
Введіть ваш вибір: 1
Введіть повне ім'я: Artem Kravchenko
Введіть номер телефону: 5656768
Введіть групу: 11
Введіть оцінки для трьох предметів через пробіл: 5 4 5
1. Додати студента
2. Видалити студента
3. Вивести студентів
4. Зберегти у файл
5. Завантажити з файлу
6. Вийти
Введіть ваш вибір: 4
1. Додати студента
2. Видалити студента
3. Вивести студентів
4. Зберегти у файл
5. Завантажити з файлу
6. Вийти
Введіть ваш вибір: █
```

Фактично витрачений час: 2 години.

### Завдання 3:

```
Індекс слова з найбільшою кількістю цифр у кожному рядку:
3
3
1
2
2
PS D:\VS Code project\C ++>
```

Фактично витрачений час: 3 години.

#### Завдання 4:

```
enter=11  
10 3  
1 2 2 3 3 3 4 5 6 7  
7  
4 5 6 7 1 2 3  
PS D:\VS Code project\C ++> |
```

Фактично витрачений час: 2 години.

#### Завдання 5:

```
enter=11  
10 3  
1 2 2 3 3 3 4 5 6 7  
7  
4 5 6 7 1 2 3  
PS D:\VS Code project\C ++> |
```

Фактично витрачений час: 1 година.



### Завдання 6:

```
K00000000  
00000000  
00000000  
00000000  
00000000  
00000000  
00000000  
00000000  
00000000
```

```
5  
1 1  
1 2  
2 1  
2 2  
3 1  
X  
K  
K  
K  
O
```

```
PS D:\VS Code project\C ++> █
```

Фактично витрачений час: 2 години.

### Завдання 7:

```
eter=m1
```

Введіть інформацію, яка буде записана у файл:

```
Hello World
```

```
Success
```

```
PS D:\VS Code project\C ++> █
```

Фактично витрачений час: 2 години.

## Завдання 8:

```
2  
HELPTOILETMENPLEASETOILET  
YES  
PS D:\VS Code project\C ++>
```

Фактично витрачений час: 40 хвилин.

## Зустріч з комадою:



**Висновок:** Виконуючи цю лабораторну роботу, я навчився працювати з бінарними та текстовими файлами, використовувати функції стандартної бібліотеки для обробки файлів, а також створювати власні бібліотеки. Це допомогло мені краще зрозуміти основи роботи з файлами та їхню роль у програмуванні.