

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Кафедра систем штучного інтелекту



Звіт

про виконання лабораторних та практичних робіт блоку № 5

На тему: «Програмування: алгоритм, програма, код. Системи числення. Двійкова система числення. Розробка та середовище розробки програми.»

з дисципліни: «Основи програмування»

до:

Практичних Робіт до блоку № 5

Виконала:

Студентка групи ШІ-13
Бобринок Ангеліна Вадимівна

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 основними поняттями роботи з файлами, включаючи бінарні та текстові файли, символи й рядкові змінні, а також засвоїти методи роботи з файлами за допомогою стандартної бібліотеки, створення та використання власних бібліотек.

1. **Теоретичні відомості:** Вступ до Роботи з Файлами:
 - Основні операції з файлами: відкриття, читання, запис, закриття
 - Робота з файловими дескрипторами
 - C-style читання з файлу та запис до файлу
 - Перевірка стану файлу: перевірка помилок, кінець файлу
 - Базові приклади читання та запису в файл
2. Символи і Рядкові Змінні:
 - Робота з char та string: основні операції і методи
 - Стрічкові літерали та екранування символів
 - Конкатенація, порівняння та пошук у рядках
3. Текстові Файли:
 - Особливості читання та запису текстових файлів
 - Обробка рядків з файлу: getline, ignore, peek
 - Форматування тексту при записі: setw, setfill, setprecision
 - Парсинг текстових файлів: розділення на слова, аналіз структури
 - Обробка помилок при роботі з файлами
4. Бінарні Файли:
 - Вступ до бінарних файлів: відмінності від текстових, приклади (великі дані, ігрові ресурси, зображення)
 - Читання та запис бінарних даних
 - Робота з позиціонуванням у файлі: seekg, seekp
 - Сериалізація об'єктів у бінарний формат
5. Стандартна бібліотека та робота з файлами:
 - Огляд стандартної бібліотеки для роботи з файлами
 - Потoki вводу/виводу: ifstream, ofstream, fstream
 - Обробка помилок при роботі з файлами
6. Створення й використання бібліотек:
 - Вступ до створення власних бібліотек у C++
 - Правила розбиття коду на header-и(.h) та source(.cpp) файли
 - Статичні проти динамічних бібліотек: переваги та використання
 - Інтерфейси бібліотек: створення, документування, версіонування
 - Використання сторонніх бібліотек у проектах

Виконання роботи:

1) Опрацювання завдання та вимог до програм та середовища:

Завдання №1

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. Надрукувати всі слова, які містять по одній цифрі.

Завдання №2

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 "Держава":

- назва;
- столиця;
- чисельність населення;
- займана площа.

Знищити всі елементи, у яких чисельність менше заданої, додати елемент після елемента із зазначеним номером.

Завдання №3

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

- 1) Скопіювати у файл F2 тільки ті рядки з F1, які починаються й закінчуються на ту саму букву.
- 2) Підрахувати кількість символів в F2.

Завдання №4

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.

Завдання №5

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

- Пуста клітинка OO
- Пішак PP
- Тура RR
- Кінь NN
- Слон BB
- Король KK
- Королева QQ

Вам дають позиції фігур на дошці (всі фігури одного кольору, кількість королів може бути > 1).

Далі йдуть QQ запитів з координатами клітинки $\{x,y\}\{x,y\}$. На кожен запит ви маєте вивести стрічку sisi - посортовані за алфавітом букви фігур, які атакують цю клітинку (пішаки атакують вниз).

У випадку, якщо на клітинці стоїть якась фігура - виведіть символ XX.

У випадку, якщо клітинку не атакують - виведіть OO.

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 №6

Реалізувати функцію створення файлу 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.

Реалізувати функцію створення файлу 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

Весна — чудова пора року: природа просинається після зимової сплячки, співають пташки, цвітуть квіточки, на вулицях з'являється все більше й більше красивих дівчат, і так важко втриматися, щоб не закохатися в якусь із них.

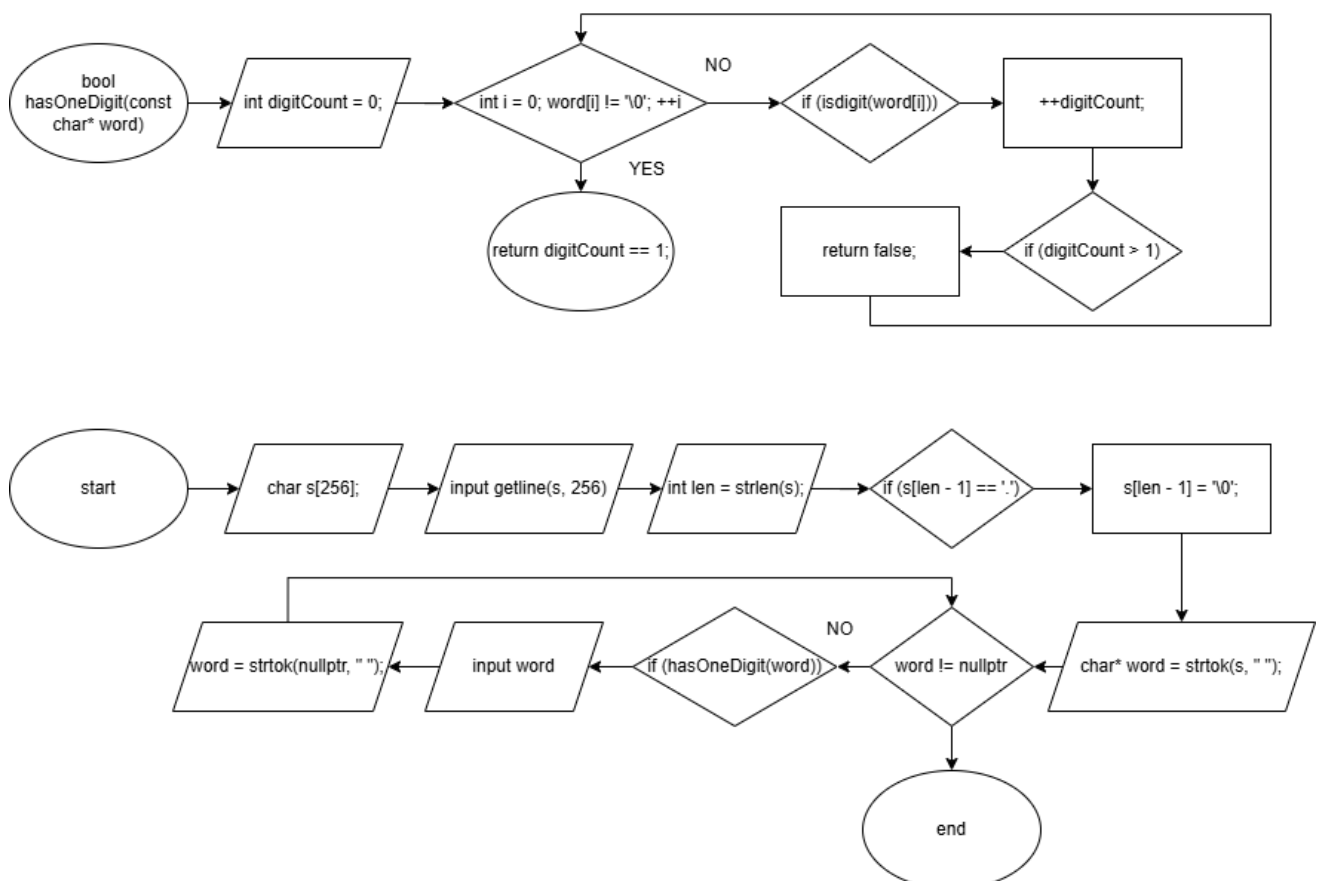
Можливо ви не повірите, але навесні може закохатися будь-хто (звичайно, якщо він не зробив цього раніше), навіть такі дивовижні створіння, як програмісти.

От і широковідомого у вузьких колах програміста Антона не минули стріли Амура — ось уже кілька тижнів йому з голови не йде одна прекрасна особа. Відверто кажучи, автор цього тексту не знає, як її звати, та це й не має ніякого значення. Коли наш закоханий іде вулицею і бачить вивіски, він машинально запам'ятовує їх, а потім думає, скільки разів можна скласти з усіх букв, які йому зустрілися, слово «весна». Або ім'я коханої. Або ще щось.

Допоможіть йому порахувати, скільки разів він зможе скласти задумане ним слово, якщо будь-яку букву можна використати не більше ніж один раз.

2) Дизайн та планована оцінка часу виконання завдань:

Завдання №1

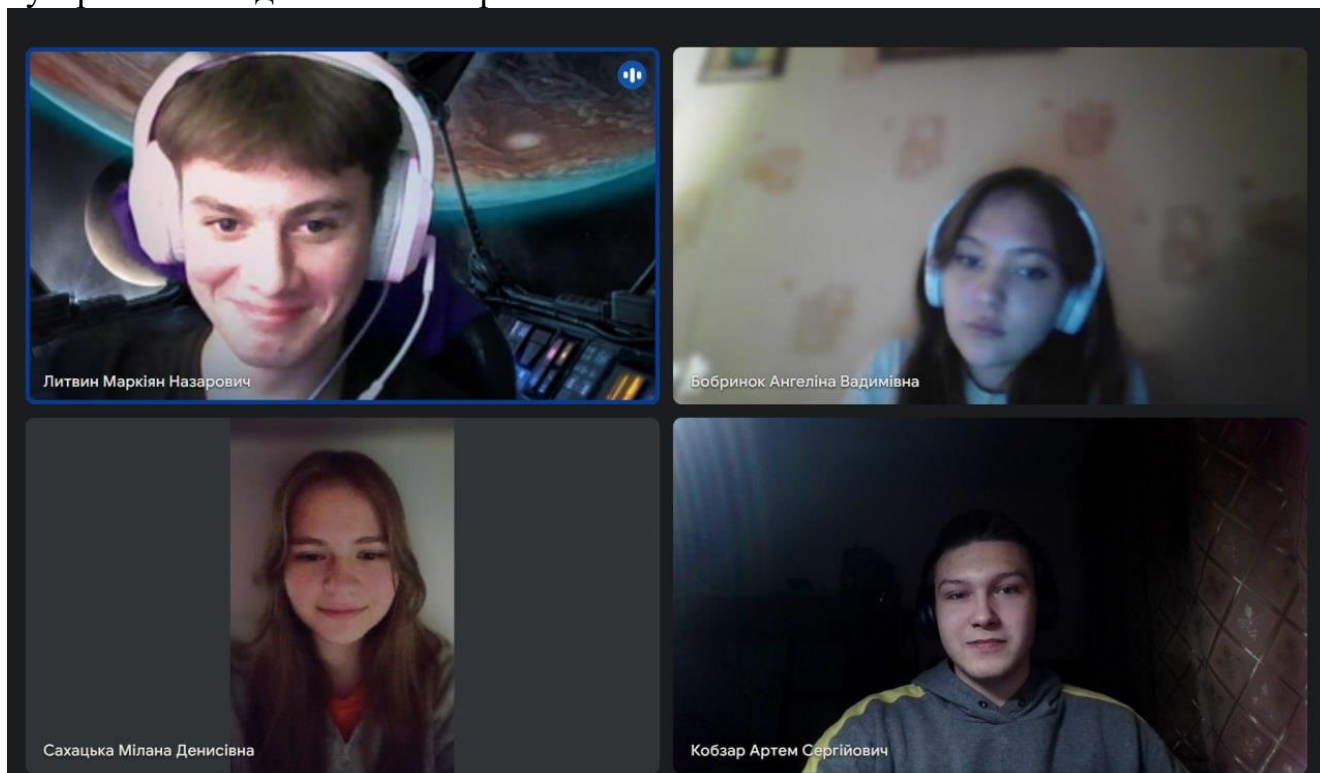


3) Конфігурація середовища до виконання завдань:

Планування роботи в notion

| epic 5 | | | | |
|------------|--|-------------|------------|--|
| Avatar | </> Task | Status | deadline | |
| Anhelina | Theory Education Activities | In progress | 05/12/2024 | |
| Anhelina | Requirements management (understand tasks) and design activities (draw flow diagrams and estimate tasks 3-9) | Not started | 05/12/2024 | |
| Anhelina | Lab# programming: VNS Lab 6 | Not started | 05/12/2024 | |
| Anhelina | Lab# programming: VNS Lab 8 | Not started | 05/12/2024 | |
| Anhelina | Lab# programming: VNS Lab 9 | Not started | 05/12/2024 | |
| Anhelina | Lab# programming: Algotester Lab 4 | Not started | 05/12/2024 | |
| Anhelina | Lab# programming: Algotester Lab 6 | Not started | 05/12/2024 | |
| Anhelina | Practice# programming: Class Practice Task | Not started | 05/12/2024 | |
| Anhelina | Practice# programming: Self Practice Task | Not started | 05/12/2024 | |
| Anhelina | Result Documentation Report and Outcomes Placement Activities (Docs and Programs on GitHub) | Not started | 05/12/2024 | |
| Anhelina | Results Evaluation and Release | Not started | 05/12/2024 | |
| + New page | | | | |

Зустріч з командою та обговорення питань



4) Код програм з посиланням на зовнішні ресурси:

Завдання №1

vns_lab_6_anhelina_bobrynok.cpp

```
1  #include <iostream>
2  #include <cstring>
3  #include <cctype>
4
5  using namespace std;
6
7  bool hasOneDigit(const char* word) {
8      int digitCount = 0;
9      for (int i = 0; word[i] != '\0'; ++i) {
10         if (isdigit(word[i])) {
11             ++digitCount;
12         }
13         if (digitCount > 1) return false;
14     }
15     return digitCount == 1;
16 }
17
18 int main() {
19     char s[256];
20
21     cout << "Введіть текст (не більше 255 символів, закінчити крапкою): ";
22     cin.getline(s, 256);
23
24     int len = strlen(s);
25     if (s[len - 1] == '.') {
26         s[len - 1] = '\0';
27     }
28
29     char* word = strtok(s, " ");
30     cout << "Слова, які містять рівно одну цифру:\n";
31
32     while (word != nullptr) {
33         if (hasOneDigit(word)) {
34             cout << word << endl;
35         }
36         word = strtok(nullptr, " ");
37     }
38
39     return 0;
40 }
```

Завдання №2

vns_lab_8_anhelina_bobrynok.cpp

```
1  #include <iostream>
2  #include <fstream>
3  #include <cstring>
4
5  using namespace std;
6
7  // Структура для країни
8  struct Country {
9      char name[50];
10     char capital[50];
11     int population;
12     double area;
13 };
14
15 // Функція для запису елементів в двійковий файл
16 void writeToFile(const char* filename, Country countries[], int count) {
17     ofstream outFile(filename, ios::binary);
18     if (!outFile) {
19         cerr << "Помилка при відкритті файлу для запису!" << endl;
20         return;
21     }
22
23     outFile.write(reinterpret_cast<char*>(&count), sizeof(count));
24     outFile.write(reinterpret_cast<char*>(countries), sizeof(Country) * count);
25
26     outFile.close();
27     cout << "Дані успішно записано у файл." << endl;
28 }
29
30 // Функція для виведення вмісту файлу
31 void printFileContents(const char* filename) {
32     ifstream inFile(filename, ios::binary);
33     if (!inFile) {
34         cerr << "Помилка при відкритті файлу для читання!" << endl;
35         return;
36     }
37
38     int count;
39     inFile.read(reinterpret_cast<char*>(&count), sizeof(count));
40
41     Country country;
42     cout << "Вміст файлу:\n";
43     for (int i = 0; i < count; ++i) {
44         inFile.read(reinterpret_cast<char*>(&country), sizeof(Country));
45         cout << "Назва: " << country.name << ", Столиця: " << country.capital
46             << ", Населення: " << country.population << ", Площа: " << country.area << endl;
47     }
48
49     inFile.close();
50 }
51
52 // Функція для знищення елементів, де чисельність населення менша за задану
53 void deleteElementsByPopulation(const char* filename, int minPopulation) {
54     ifstream inFile(filename, ios::binary);
55     if (!inFile) {
56         cerr << "Помилка при відкритті файлу для читання!" << endl;
57         return;
58     }
59 }
```



```

60     int count;
61     inFile.read(reinterpret_cast<char*>(&count), sizeof(count));
62
63     Country* countries = new Country[count];
64     for (int i = 0; i < count; ++i) {
65         inFile.read(reinterpret_cast<char*>(&countries[i]), sizeof(Country));
66     }
67     inFile.close();
68
69     // Визначення нового розміру
70     int newCount = 0;
71     for (int i = 0; i < count; ++i) {
72         if (countries[i].population >= minPopulation) {
73             countries[newCount++] = countries[i];
74         }
75     }
76
77     // Перезаписування файлу з новими даними
78     writeToFile(filename, countries, newCount);
79
80     delete[] countries;
81     cout << "Елементи з чисельністю населення менше " << minPopulation << " були видалені." << endl;
82 }
83
84 // Функція для додавання нового елемента після заданого індексу
85 void addElementAfterIndex(const char* filename, int index, Country newCountry) {
86     ifstream inFile(filename, ios::binary);
87     if (!inFile) {
88         cerr << "Помилка при відкритті файлу для читання!" << endl;
89         return;
90     }
91
92     int count;
93     inFile.read(reinterpret_cast<char*>(&count), sizeof(count));
94
95     Country* countries = new Country[count + 1];
96
97     // Читання всіх елементів файлу
98     for (int i = 0; i < count; ++i) {
99         inFile.read(reinterpret_cast<char*>(&countries[i]), sizeof(Country));
100     }
101     inFile.close();
102
103     // Додавання нового елемента після вказаного індексу
104     for (int i = count; i > index; --i) {
105         countries[i] = countries[i - 1];
106     }
107     countries[index + 1] = newCountry;
108
109     // Перезаписування файлу з новими даними
110     writeToFile(filename, countries, count + 1);
111
112     delete[] countries;
113     cout << "Елемент успішно додано після індексу " << index << "." << endl;

```

```

114     }
115
116     int main() {
117         const char* filename = "countries.dat";
118
119         // Приклад країн
120         Country countries[] = {
121             {"Ukraine", "Kyiv", 4105267, 603550},
122             {"Poland", "Warsaw", 38386000, 312696},
123             {"Germany", "Berlin", 83019200, 357022},
124             {"USA", "Washington", 331002651, 9833517}
125         };
126
127
128         writeToFile(filename, countries, 4);
129
130         printFileContents(filename);
131
132         deleteElementsByPopulation(filename, 40000000);
133
134         Country newCountry = {"France", "Paris", 67081000, 551695};
135         addElementAfterIndex(filename, 1, newCountry);
136
137         printFileContents(filename);
138
139         return 0;
140     }
141

```

Завдання №3

vns_lab_9_anhelina_bobrynok.cpp

```

1  #include <iostream>
2  #include <fstream>
3  #include <cstring>
4
5  using namespace std;
6
7  // Функція для перевірки, чи рядок починається і закінчується на одну й ту саму букву
8  bool startsAndEndsWithSameLetter(const string& line) {
9      if (line.empty()) return false;
10     char firstChar = line[0];
11     char lastChar = line[line.size() - 1];
12     return (tolower(firstChar) == tolower(lastChar));
13 }
14
15 int main() {
16     const char* inputFilename = "F1.txt";
17     const char* outputFilename = "F2.txt";
18
19     // Відкриваємо файл F1 для читання
20     ifstream inputFile(inputFilename);
21     if (!inputFile) {
22         cerr << "Помилка при відкритті файлу " << inputFilename << endl;
23         return 1;
24     }
25
26     // Відкриваємо файл F2 для запису
27     ofstream outputFile(outputFilename);
28     if (!outputFile) {
29         cerr << "Помилка при відкритті файлу " << outputFilename << endl;
30         return 1;
31     }

```

```

33     string line;
34     int totalChars = 0; // Лічильник символів у F2
35
36     // Читання кожного рядка з файлу F1
37     while (getline(inputFile, line)) {
38         if (startsAndEndsWithSameLetter(line)) {
39             outputFile << line << endl;
40             totalChars += line.size();
41         }
42     }
43
44     inputFile.close();
45     outputFile.close();
46
47     // Виведення кількості символів у F2
48     cout << "Кількість символів у файлі F2: " << totalChars << endl;
49
50     return 0;
51 }

```

Завдання №4

algotester_lab_4_v1_anhelina_bobrynok.cpp

```

1  #include <iostream>
2  #include <vector>
3  #include <set>
4  #include <algorithm>
5  #include <iterator>
6
7  using namespace std;
8
9  int main() {
10     int N, K;
11     cin >> N >> K;
12
13     vector<int> a(N);
14     for (int i = 0; i < N; i++) {
15         cin >> a[i];
16     }
17
18     set<int> unique_set(a.begin(), a.end());
19
20     vector<int> sorted(unique_set.begin(), unique_set.end());
21
22     cout << sorted.size() << endl;
23
24     rotate(sorted.begin(), sorted.begin() + K % sorted.size(), sorted.end());
25
26     for (int x : sorted) {
27         cout << x << " ";
28     }
29     cout << endl;
30
31     return 0;
32 }

```

algotester_lab_4_v2_anhelina_bobrynok.cpp

```

1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      int N, K;
7      cin >> N >> K;
8
9      int a[N];
10     bool unique[1000001] = { false };
11     int unique_count = 0;
12
13     for (int i = 0; i < N; i++) {
14         int num;
15         cin >> num;
16         if (!unique[num]) {
17             unique[num] = true;
18             a[unique_count++] = num;
19         }
20     }
21
22     for (int i = 0; i < unique_count - 1; i++) {
23         for (int j = 0; j < unique_count - i - 1; j++) {
24             if (a[j] > a[j + 1]) {
25                 int temp = a[j];
26                 a[j] = a[j + 1];
27                 a[j + 1] = temp;
28             }
29         }
30     }
31
32     cout << unique_count << endl;
33
34     K = K % unique_count;
35
36     for (int i = 0; i < unique_count; i++) {
37         cout << a[(i + K) % unique_count] << " ";
38     }
39     cout << endl;
40
41     return 0;
42 }
43

```

Завдання №5

algotester_lab_6_anhelina_bobrynok .cpp

```
1  #include <iostream>
2  #include <vector>
3  #include <cmath>
4  #include <string>
5  #include <algorithm>
6  using namespace std;
7
8  struct point
9  {
10     char name;
11     int x;
12     int y;
13 };
14
15 bool x_case(vector<point> &v, int x, int y)
16 {
17     for (auto t : v)
18         if (t.x == x && t.y == y)
19             return true;
20     return false;
21 }
22
23 char check(point p, int x, int y)
24 {
25     switch (p.name)
26     {
27     case 'P':
28         if (x - p.x == 1 && abs(y - p.y) == 1)
29             return 'P';
30         break;
31     case 'R':
32         if (p.x == x || p.y == y)
33             return 'R';
34         break;
35     case 'N':
36         if ((abs(p.x - x) == 1 && abs(p.y - y) == 2) || (abs(p.x - x) == 2 && abs(p.y - y) == 1))
37             return 'N';
38         break;
39     case 'B':
40         if (abs(p.x - x) == abs(p.y - y))
41             return 'B';
42         break;
43     case 'K':
44         if (abs(p.x - x) <= 1 && abs(p.y - y) <= 1)
45             return 'K';
46         break;
47     case 'Q':
48         if ((p.x == x || p.y == y) || (abs(p.x - x) == abs(p.y - y)))
49             return 'Q';
50         break;
51     default:
52         break;
53     }
54     return 'O';
55 }
56
57 int main()
58 {
59     vector<string> s(8);
60     vector<point> v;
```

```

61     for (int i = 0; i < 8; i++)
62     {
63         getline(cin, s[i]);
64         for (int j = 0; j < 8; j++)
65             if (s[i][j] != '0')
66                 v.push_back({s[i][j], i, j});
67     }
68
69     int q;
70     cin >> q;
71     cin.ignore();
72     vector<string> answers;
73
74     for (int i = 0; i < q; i++)
75     {
76         int x, y;
77         cin >> x >> y;
78         x = x - 1;
79         y = y - 1;
80
81         string answer;
82         if (x_case(v, x, y))
83             answer = "X";
84         else
85         {
86             for (auto p : v)
87             {
88                 const char el = check(p, x, y);
89                 if (el != '0')
90                     answer.push_back(el);
91             }
92             if (answer.size() == 0)
93                 answer.push_back('0');
94         }
95         sort(answer.begin(), answer.end());
96         answer.erase(unique(answer.begin(), answer.end()), answer.end());
97         answers.push_back(answer);
98     }
99
100     for (auto a : answers)
101         cout << a << endl;
102
103     return 0;
104 }

```

Завдання №6

practice_work_task_1_anhelina_bobrynok.cpp

```

1  #include <iostream>
2  #include <fstream>
3  #include <string>
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         return Failure;
12     }
13
14     ofstream file(name, ios::out);
15     if (!file) {
16         return Failure;
17     }
18
19     file << content;
20     if (!file.good()) {
21         return Failure;
22     }
23
24     file.close();
25     if (!file.good()) {
26         return Failure;
27     }
28
29     return Success;
30 }
31
32 int main() {
33     string filename;
34     string content;
35
36     cout << "Введіть ім'я файлу: ";
37     getline(cin, filename);
38
39     cout << "Введіть вміст для запису у файл: ";
40     getline(cin, content);
41
42     FileOpResult result = write_to_file(filename.c_str(), content.c_str());
43     if (result == Success) {
44         cout << "Файл успішно створено і записано.\n";
45     } else {
46         cout << "Сталася помилка при записі у файл.\n";
47     }
48
49     return 0;
50 }

```

practice_work_task_2_anhelina_bobrynok.cpp

```

1  #include <iostream>
2  #include <fstream>
3  #include <string>
4
5  using namespace std;
6
7  enum FileOpResult { Success, Failure };
8
9  FileOpResult copy_file(const char *file_from, const char *file_to) {
10     if (!file_from || !file_to) {
11         return Failure;
12     }
13
14     ifstream input(file_from, ios::in);
15     if (!input) {
16         return Failure;
17     }
18
19     ofstream output(file_to, ios::out);
20     if (!output) {
21         return Failure;
22     }
23
24     string line;
25     while (getline(input, line)) {
26         output << line << "\n";
27     }
28
29     if (!output.good()) {
30         return Failure;
31     }
32
33     input.close();
34     output.close();
35
36     if (!input.good() || !output.good()) {
37         return Failure;
38     }
39
40     return Success;
41 }
42
43 int main() {
44     string file_from, file_to;
45
46     cout << "Введіть ім'я вихідного файлу: ";
47     getline(cin, file_from);
48
49     cout << "Введіть ім'я цільового файлу: ";
50     getline(cin, file_to);
51
52     FileOpResult result = copy_file(file_from.c_str(), file_to.c_str());
53     if (result == Success) {
54         cout << "Файл успішно скопійовано.\n";
55     } else {
56         cout << "Сталася помилка при копіюванні файлу.\n";
57     }
58
59     return 0;
60 }

```


Завдання №7

self_practice_work_anhelina_bobrynok.cpp

```
1  #include <iostream>
2  #include <string>
3  #include <map>
4  #include <algorithm>
5  #include <climits>
6
7  using namespace std;
8
9  map<char, int> countLetters(const string& s) {
10     map<char, int> freq;
11     for (char c : s) {
12         ++freq[c];
13     }
14     return freq;
15 }
16
17 int main() {
18     string targetWord;
19     cin >> targetWord;
20
21     int n;
22     cin >> n;
23
24     map<char, int> totalLetters;
25
26     for (int i = 0; i < n; ++i) {
27         string sign;
28         cin >> sign;
29         for (char c : sign) {
30             ++totalLetters[c];
31         }
32     }
33
34     map<char, int> targetFreq = countLetters(targetWord);
35
36     int minCount = INT_MAX;
37     for (auto& [letter, count] : targetFreq) {
38         if (totalLetters[letter] == 0) {
39             minCount = 0;
40             break;
41         }
42         minCount = min(minCount, totalLetters[letter] / count);
43     }
44
45     cout << minCount << endl;
46     return 0;
47 }
```

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

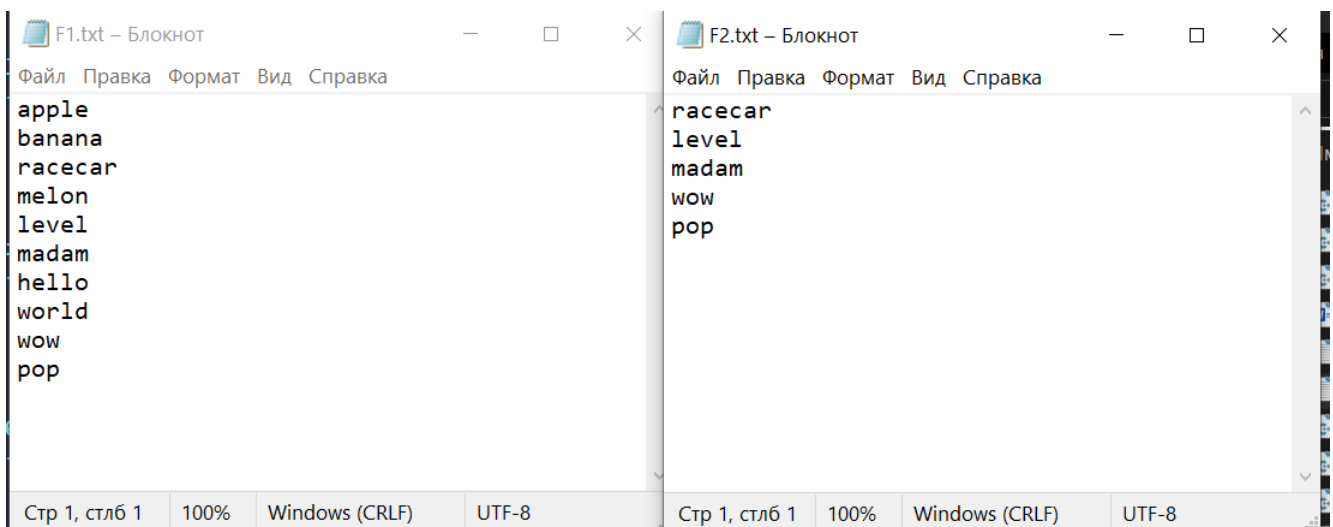
Завдання №1

Введіть текст (не більше 255 символів, закінчити крапкою): hello world.
Слова, які містять рівно одну цифру:
w0rld

Завдання №2

```
Дані успішно записано у файл.  
Вміст файлу:  
Назва: Ukraine, Столиця: Kyiv, Населення: 4105267, Площа: 603550  
Назва: Poland, Столиця: Warsaw, Населення: 38386000, Площа: 312696  
Назва: Germany, Столиця: Berlin, Населення: 83019200, Площа: 357022  
Назва: USA, Столиця: Washington, Населення: 331002651, Площа: 9.83352e+06  
Дані успішно записано у файл.  
Елементи з чисельністю населення менше 40000000 були видалені.  
Дані успішно записано у файл.  
Елемент успішно додано після індексу 1.  
Вміст файлу:  
Назва: Germany, Столиця: Berlin, Населення: 83019200, Площа: 357022  
Назва: USA, Столиця: Washington, Населення: 331002651, Площа: 9.83352e+06  
Назва: France, Столиця: Paris, Населення: 67081000, Площа: 551695
```

Завдання №3



Завдання №4

| Створено | Компілятор | Результат | Час (сек.) | Пам'ять (МіБ) | Дії |
|----------------------|------------|------------|------------|---------------|--------------------------|
| декілька секунд тому | C++ 23 | Зараховано | 0.003 | 1.438 | Перегляд |

| Створено | Компілятор | Результат | Час (сек.) | Пам'ять (МіБ) | Дії |
|----------------------|------------|------------|------------|---------------|--------------------------|
| декілька секунд тому | C++ 23 | Зараховано | 0.004 | 1.254 | Перегляд |

Завдання №5

| Створено | Компілятор | Результат | Час (сек.) | Пам'ять (МіБ) | Дії |
|----------------------|------------|------------|------------|---------------|--------------------------|
| декілька секунд тому | C++ 23 | Зараховано | 0.003 | 1.254 | Перегляд |

Завдання №6

```
Введіть ім'я файлу: hello world
Введіть вміст для запису у файл: hello world!
Файл успішно створено і записано.
```

```
Введіть ім'я вихідного файлу: hi
Введіть ім'я цільового файлу: bye
Сталася помилка при копіюванні файлу.
```

Завдання №7

| Створено | Компілятор | Результат | Час (сек.) | Пам'ять (МіБ) | Дії |
|----------------------|------------|------------|------------|---------------|--------------------------|
| декілька секунд тому | C++ 23 | Зараховано | 0.003 | 1.207 | Перегляд |

Висновки: Завдяки цій лабораторній роботі я мала змогу ознайомитися з основними поняттями роботи з файлами, включаючи бінарні та текстові файли, символи й рядкові змінні, а також засвоїти методи роботи з файлами за допомогою стандартної бібліотеки, створення та використання власних бібліотек.