**Індивідуальна робота Номер 1**
**Надененко Олексій МФ-21**

**Постановка задачі:**
Олімпіада з програмування. Є два текстові файли. У першому міститься інформація про країни-учасниці та подані ними командами у вигляді набору рядків, у кожному з яких назва країни та перелік назв команд цієї країни. У другому – перелік назв команд у порядку зайнятих ними місць (передбачається, що немає команд із однаковими назвами). Потрібно створити список списків: список назв країн, упорядкований за абеткою, і для кожної країни – список команд із зазначенням зайнятого ними місця у порядку зростання. Забезпечити операції додавання, видалення, редагування та розумного пошуку інформації, виведення інформації в розумному вигляді...

Ця програма надає повноцінний інструмент для роботи з країнами та командами, дозволяючи завантажувати інформацію, впорядковувати її, виконувати додавання, видалення, редагування та пошук. Основні моменти реалізації:

**Структура даних:**
Використовуються однозв'язні списки:
Для країн (Country).
Для команд всередині кожної країни (Team).
Це забезпечує гнучкість у додаванні та видаленні елементів без необхідності перерозподілу пам'яті.

**Функції програми:**
Додавання країн і команд із упорядкуванням за абеткою.
Завантаження даних із файлів:
Список країн із командами.
Рейтинг команд.
Редагування назви команди.
Видалення країн і команд із очищенням пам'яті.
Пошук команди за назвою.
Виведення інформації у структурованому вигляді.
Звільнення пам'яті при завершенні роботи.

**Файли вхідних даних:**
**countries.txt:** Інформація про країни та їхні команди у форматі:
makefile
Ukraine: TeamA, TeamB, TeamC
USA: TeamX, TeamY
**ranking.txt:** Рейтинг команд за зайнятими місцями:
TeamX
TeamB
TeamA

**Меню для інтерактивного управління:**
Користувач може виконувати операції безпосередньо під час роботи програми.
**Очищення пам'яті:**
Звільнення пам'яті здійснюється як для списків країн, так і для списків команд, запобігаючи витокам пам'яті.
Цей код демонструє приклад систематизованого підходу до задачі, забезпечуючи необхідну функціональність із мінімальною кількістю витрат ресурсів.

**Обґрунтування вибору структури даних**
Динамічність: Однозв’язні списки дозволяють гнучко додавати/видаляти країни й команди без попереднього резервування пам’яті, що важливо при невідомій кількості елементів.

Ефективність операцій:
Додавання: O(n)O(n)O(n) для пошуку позиції та O(1)O(1)O(1) для вставки.
Видалення: O(n)O(n)O(n) без переміщення інших елементів, як у масивах.

Вкладеність: Список списків (країни → команди) легко моделюється через однозв’язні структури.

Впорядкування: Країни автоматично впорядковуються під час вставки, без потреби повторного сортування.
Це спрощує реалізацію, ефективно використовує пам’ять і забезпечує швидку обробку.

**Опис Програми**

```
#include <iostream>
#include <fstream>
#include <string>

using namespace std;
```

**Структура для збереження інформації про команду**

```
struct Team {
    string name;   //Назва команди
    int place;     //Зайняте місце (або -1, якщо не визначено)
    Team* next;   // Вказівник на наступну команду
};
```

**Структура для збереження інформації про країну**

```
struct Country {
    string name;  // Назва країни
    Team* teams;  // Вказівник на список команд цієї країни
    Country* next; // Вказівник на наступну країну
};
```

**Глобальна змінна для збереження голови списку країн**

```
Country* head = nullptr;
```

**Додає країну до списку країн в алфавітному порядку**

```
void add(const string& name) {
    if (name.empty()) {
        cout << "Назва країни не може бути порожньою.\n";
        return;
    }

    Country* newCountry = new Country{name, nullptr, nullptr};
    if (!head || head->name > name) {
        newCountry->next = head;
        head = newCountry;
    } else {
        Country* current = head;
        while (current->next && current->next->name < name) {
```

```
            current = current->next;
        }
        newCountry->next = current->next;
        current->next = newCountry;
    }
}
```

**Додає команду до вказаної країни; місце команди опційне**

```
void addTeam(const string& country, const string& name, int place = -1) {
    if (country.empty() || name.empty()) {
        cout << "Назва країни або команди не може бути порожньою.\n";
        return;
    }

    Country* current = head;
    while (current && current->name != country) {
        current = current->next;
    }
    if (!current) {
        cout << "Країну " << country << " не знайдено.\n";
        return;
    }

    Team* newTeam = new Team{name, place, nullptr};
    if (!current->teams || (place != -1 && current->teams->place > place)) {
        newTeam->next = current->teams;
        current->teams = newTeam;
    } else {
        Team* teamCurrent = current->teams;
        while (teamCurrent->next && (place == -1 || teamCurrent->next->place < place)) {
            teamCurrent = teamCurrent->next;
        }
        newTeam->next = teamCurrent->next;
        teamCurrent->next = newTeam;
    }
}
```

**Видаляє зайві пробіли з рядка**

```
string trim(const string& str) {
    size_t first = str.find_first_not_of(" \t");
    if (first == string::npos) return "";
    size_t last = str.find_last_not_of(" \t");
    return str.substr(first, (last - first + 1));
}
```

**Завантажує інформацію про країни та їхні команди з текстового файлу**

```
void load(const string& filename) {
    try {
        ifstream file(filename);
        if (!file) {
            throw runtime_error("Не вдалося відкрити файл " + filename);
        }

        string line;
        while (getline(file, line)) {
```

```
            size_t colonPos = line.find(':');
            if (colonPos == string::npos) continue;

            string countryName = trim(line.substr(0, colonPos));
            string teamsList = line.substr(colonPos + 1);
            add(countryName);

            size_t startPos = 0, commaPos;
            while ((commaPos = teamsList.find(',', startPos)) != string::npos) {
                string teamName = trim(teamsList.substr(startPos, commaPos - startPos));
                addTeam(countryName, teamName);
                startPos = commaPos + 1;
            }
            string teamName = trim(teamsList.substr(startPos));
            addTeam(countryName, teamName);
        }

        file.close();
    } catch (const exception& e) {
        cout << e.what() << endl;
    }
}
```

**Завантажує рейтинг команд з файлу та оновлює інформацію про зайняті місця**

```
void loadRankings(const string& filename) {
    try {
        ifstream file(filename);
        if (!file) {
            throw runtime_error("Не вдалося відкрити файл " + filename);
        }

        string teamName;
        int place = 1;
        while (getline(file, teamName)) {
            Country* currentCountry = head;
            bool found = false;

            while (currentCountry) {
                Team* currentTeam = currentCountry->teams;
                while (currentTeam) {
                    if (currentTeam->name == teamName) {
                        currentTeam->place = place++;
                        found = true;
                        break;
                    }
                    currentTeam = currentTeam->next;
                }
                if (found) break;
                currentCountry = currentCountry->next;
            }

            if (!found) {
                cout << "Команду " << teamName << " не знайдено у жодній країні." << endl;
            }
        }
```

```
        file.close();
    } catch (const exception& e) {
        cout << e.what() << endl;
    }
}
```

**Видаляє країну та всі її команди зі списку**

```
void removeCountry(const string& name) {
    if (name.empty()) {
        cout << "Назва країни не може бути порожньою.\n";
        return;
    }

    Country* current = head;
    Country* previous = nullptr;

    while (current && current->name != name) {
        previous = current;
        current = current->next;
    }

    if (!current) {
        cout << "Країну " << name << " не знайдено.\n";
        return;
    }

    Team* teamCurrent = current->teams;
    while (teamCurrent) {
        Team* tempTeam = teamCurrent;
        teamCurrent = teamCurrent->next;
        delete tempTeam;
    }

    if (previous) {
        previous->next = current->next;
    } else {
        head = current->next;
    }
    delete current;

    cout << "Країну " << name << " видалено.\n";
}
```

**Видаляє команду зі списку**

```
void removeTeam(const string& name) {
    if (name.empty()) {
        cout << "Назва команди не може бути порожньою.\n";
        return;
    }

    Country* currentCountry = head;

    while (currentCountry) {
        Team* currentTeam = currentCountry->teams;
        Team* previousTeam = nullptr;
```

```
        while (currentTeam && currentTeam->name != name) {
            previousTeam = currentTeam;
            currentTeam = currentTeam->next;
        }

        if (currentTeam) {
            if (previousTeam) {
                previousTeam->next = currentTeam->next;
            } else {
                currentCountry->teams = currentTeam->next;
            }
            delete currentTeam;
            cout << "Команду " << name << " видалено.\n";
            return;
        }
        currentCountry = currentCountry->next;
    }

    cout << "Команду " << name << " не знайдено.\n";
}
```

**Виводить список країн та їхніх команд з місцями**

```
void show() {
    Country* currentCountry = head;
    while (currentCountry) {
        cout << "Країна: " << currentCountry->name << endl;
        Team* currentTeam = currentCountry->teams;
        while (currentTeam) {
            cout << "  Команда: " << currentTeam->name
                 << ", Місце: " << (currentTeam->place == -1 ? "не визначено" :
to_string(currentTeam->place)) << endl;
            currentTeam = currentTeam->next;
        }
        currentCountry = currentCountry->next;
    }
}
```

**Очищає всі дані зі списку**

```
void clear() {
    while (head) {
        Country* tempCountry = head;
        head = head->next;
        while (tempCountry->teams) {
            Team* tempTeam = tempCountry->teams;
            tempCountry->teams = tempCountry->teams->next;
            delete tempTeam;
        }
        delete tempCountry;
    }
}
```

**Шукає команду за назвою**

```
void search(const string& name) {
    if (name.empty()) {
        cout << "Назва команди не може бути порожньою.\n";
        return;
```

```
    }

    Country* currentCountry = head;
    while (currentCountry) {
        Team* currentTeam = currentCountry->teams;
        while (currentTeam) {
            if (currentTeam->name == name) {
                cout << "Команда " << name << " належить країні " << currentCountry->name
                    << " та зайняла місце " << (currentTeam->place == -1 ? "не визначено" :
to_string(currentTeam->place)) << endl;
                return;
            }
            currentTeam = currentTeam->next;
        }
        currentCountry = currentCountry->next;
    }
    cout << "Команду " << name << " не знайдено." << endl;
}
```

**Перейменовує команду**

```
void renameTeam(const string& oldName, const string& newName) {
    if (oldName.empty() || newName.empty()) {
        cout << "Назва команди не може бути порожньою.\n";
        return;
    }

    Country* currentCountry = head;
    while (currentCountry) {
        Team* currentTeam = currentCountry->teams;
        while (currentTeam) {
            if (currentTeam->name == oldName) {
                currentTeam->name = newName;
                cout << "Назву команди " << oldName << " змінено на " << newName << "." <<
endl;
                return;
            }
            currentTeam = currentTeam->next;
        }
        currentCountry = currentCountry->next;
    }
    cout << "Команду " << oldName << " не знайдено." << endl;
}
```

**Виводить меню програми**

```
void menu() {
    cout << "\n=== Меню ===\n";
    cout << "1. Додати країну\n";
    cout << "2. Додати команду\n";
    cout << "3. Змінити назву\n";
    cout << "4. Видалити країну\n";
    cout << "5. Видалити команду\n";
    cout << "6. Знайти команду\n";
    cout << "7. Показати всі дані\n";
    cout << "8. Вийти\n";
    cout << "Виберіть опцію: ";
}
```

**Основна функція програми**

```
int main() {
    load("countries.txt");        Завантажує дані про країни та команди
    loadRankings("ranking.txt");  Завантажує рейтинги команд

    int choice;
    string countryName, teamName, newName;

    while (true) {
        menu();
        cin >> choice;

        switch (choice) {
            case 1:
                cout << "Введіть назву країни: ";
                cin >> ws;
                getline(cin, countryName);
                add(countryName);
                break;
            case 2:
                cout << "Введіть назву країни: ";
                cin >> ws;
                getline(cin, countryName);
                cout << "Введіть назву команди: ";
                getline(cin, teamName);
                addTeam(countryName, teamName);
                break;
            case 3:
                cout << "Введіть стару назву: ";
                cin >> ws;
                getline(cin, teamName);
                cout << "Введіть нову назву: ";
                getline(cin, newName);
                renameTeam(teamName, newName);
                break;
            case 4:
                cout << "Введіть назву країни: ";
                cin >> ws;
                getline(cin, countryName);
                removeCountry(countryName);
                break;
            case 5:
                cout << "Введіть назву команди: ";
                cin >> ws;
                getline(cin, teamName);
                removeTeam(teamName);
                break;
            case 6:
                cout << "Введіть назву команди: ";
                cin >> ws;
                getline(cin, teamName);
                search(teamName);
                break;
            case 7:
                show();
```

```
            break;
        case 8:
            cout << "Вихід з програми.\n";
            clear();
            return 0;
        default:
            cout << "Невірний вибір. Спробуйте ще раз.\n";
    }
  }
}
```

**Код программы**:

```
#include <iostream>
#include <fstream>
#include <string>
#include <stdexcept>

using namespace std;

struct Team {
  string name;
  int place;
  Team* next;
};

struct Country {
  string name;
  Team* teams;
  Country* next;
};

Country* head = nullptr;

void add(const string& name) {
  if (name.empty()) {
    cout << "Назва країни не може бути порожньою.\n";
    return;
  }

  Country* newCountry = new Country{name, nullptr, nullptr};
  if (!head || head->name > name) {
    newCountry->next = head;
    head = newCountry;
  } else {
    Country* current = head;
    while (current->next && current->next->name < name) {
      current = current->next;
    }
    newCountry->next = current->next;
    current->next = newCountry;
  }
}

void addTeam(const string& country, const string& name, int place = -1) {
  if (country.empty() || name.empty()) {
```

```
        cout << "Назва країни або команди не може бути порожньою.\n";
        return;
    }

    Country* current = head;
    while (current && current->name != country) {
        current = current->next;
    }
    if (!current) {
        cout << "Країну " << country << " не знайдено.\n";
        return;
    }

    Team* newTeam = new Team{name, place, nullptr};
    if (!current->teams || (place != -1 && current->teams->place > place)) {
        newTeam->next = current->teams;
        current->teams = newTeam;
    } else {
        Team* teamCurrent = current->teams;
        while (teamCurrent->next && (place == -1 || teamCurrent->next->place < place)) {
            teamCurrent = teamCurrent->next;
        }
        newTeam->next = teamCurrent->next;
        teamCurrent->next = newTeam;
    }
}

string trim(const string& str) {
    size_t first = str.find_first_not_of(" \t");
    if (first == string::npos) return "";
    size_t last = str.find_last_not_of(" \t");
    return str.substr(first, (last - first + 1));
}

void load(const string& filename) {
    try {
        ifstream file(filename);
        if (!file) {
            throw runtime_error("Не вдалося відкрити файл " + filename);
        }

        string line;
        while (getline(file, line)) {
            size_t colonPos = line.find(':');
            if (colonPos == string::npos) continue;

            string countryName = trim(line.substr(0, colonPos));
            string teamsList = line.substr(colonPos + 1);
            add(countryName);

            size_t startPos = 0, commaPos;
            while ((commaPos = teamsList.find(',', startPos)) != string::npos) {
                string teamName = trim(teamsList.substr(startPos, commaPos - startPos));
                addTeam(countryName, teamName);
                startPos = commaPos + 1;
            }
```

```
            string teamName = trim(teamsList.substr(startPos));
            addTeam(countryName, teamName);
        }

        file.close();
    } catch (const exception& e) {
        cout << e.what() << endl;
    }
}

void loadRankings(const string& filename) {
    try {
        ifstream file(filename);
        if (!file) {
            throw runtime_error("Не вдалося відкрити файл " + filename);
        }

        string teamName;
        int place = 1;
        while (getline(file, teamName)) {
            Country* currentCountry = head;
            bool found = false;

            while (currentCountry) {
                Team* currentTeam = currentCountry->teams;
                while (currentTeam) {
                    if (currentTeam->name == teamName) {
                        currentTeam->place = place++;
                        found = true;
                        break;
                    }
                    currentTeam = currentTeam->next;
                }
                if (found) break;
                currentCountry = currentCountry->next;
            }

            if (!found) {
                cout << "Команду " << teamName << " не знайдено у жодній країні." << endl;
            }
        }

        file.close();
    } catch (const exception& e) {
        cout << e.what() << endl;
    }
}

void removeCountry(const string& name) {
    if (name.empty()) {
        cout << "Назва країни не може бути порожньою.\n";
        return;
    }

    Country* current = head;
    Country* previous = nullptr;
```

```
    while (current && current->name != name) {
        previous = current;
        current = current->next;
    }

    if (!current) {
        cout << "Країну " << name << " не знайдено.\n";
        return;
    }

    Team* teamCurrent = current->teams;
    while (teamCurrent) {
        Team* tempTeam = teamCurrent;
        teamCurrent = teamCurrent->next;
        delete tempTeam;
    }

    if (previous) {
        previous->next = current->next;
    } else {
        head = current->next;
    }
    delete current;

    cout << "Країну " << name << " видалено.\n";
}

void removeTeam(const string& name) {
    if (name.empty()) {
        cout << "Назва команди не може бути порожньою.\n";
        return;
    }

    Country* currentCountry = head;

    while (currentCountry) {
        Team* currentTeam = currentCountry->teams;
        Team* previousTeam = nullptr;

        while (currentTeam && currentTeam->name != name) {
            previousTeam = currentTeam;
            currentTeam = currentTeam->next;
        }

        if (currentTeam) {
            if (previousTeam) {
                previousTeam->next = currentTeam->next;
            } else {
                currentCountry->teams = currentTeam->next;
            }
            delete currentTeam;
            cout << "Команду " << name << " видалено.\n";
            return;
        }
        currentCountry = currentCountry->next;
```

```
    }

    cout << "Команду " << name << " не знайдено.\n";
}

void show() {
    Country* currentCountry = head;
    while (currentCountry) {
        cout << "Країна: " << currentCountry->name << endl;
        Team* currentTeam = currentCountry->teams;
        while (currentTeam) {
            cout << "  Команда: " << currentTeam->name
                 << ", Місце: " << (currentTeam->place == -1 ? "не визначено" :
to_string(currentTeam->place)) << endl;
            currentTeam = currentTeam->next;
        }
        currentCountry = currentCountry->next;
    }
}

void clear() {
    while (head) {
        Country* tempCountry = head;
        head = head->next;
        while (tempCountry->teams) {
            Team* tempTeam = tempCountry->teams;
            tempCountry->teams = tempCountry->teams->next;
            delete tempTeam;
        }
        delete tempCountry;
    }
}

void search(const string& name) {
    if (name.empty()) {
        cout << "Назва команди не може бути порожньою.\n";
        return;
    }

    Country* currentCountry = head;
    while (currentCountry) {
        Team* currentTeam = currentCountry->teams;
        while (currentTeam) {
            if (currentTeam->name == name) {
                cout << "Команда " << name << " належить країні " << currentCountry->name
                     << " та зайняла місце " << (currentTeam->place == -1 ? "не визначено" :
to_string(currentTeam->place)) << endl;
                return;
            }
            currentTeam = currentTeam->next;
        }
        currentCountry = currentCountry->next;
    }
    cout << "Команду " << name << " не знайдено." << endl;
}
```

```
void renameTeam(const string& oldName, const string& newName) {
    if (oldName.empty() || newName.empty()) {
        cout << "Назва команди не може бути порожньою.\n";
        return;
    }

    Country* currentCountry = head;
    while (currentCountry) {
        Team* currentTeam = currentCountry->teams;
        while (currentTeam) {
            if (currentTeam->name == oldName) {
                currentTeam->name = newName;
                cout << "Назву команди " << oldName << " змінено на " << newName << "." <<
endl;
                return;
            }
            currentTeam = currentTeam->next;
        }
        currentCountry = currentCountry->next;
    }
    cout << "Команду " << oldName << " не знайдено." << endl;
}

void menu() {
    cout << "\n=== Меню ===\n";
    cout << "1. Додати країну\n";
    cout << "2. Додати команду\n";
    cout << "3. Змінити назву\n";
    cout << "4. Видалити країну\n";
    cout << "5. Видалити команду\n";
    cout << "6. Знайти команду\n";
    cout << "7. Показати всі дані\n";
    cout << "8. Вийти\n";
    cout << "Виберіть опцію: ";
}

int main() {
    load("countries.txt");
    loadRankings("ranking.txt");

    int choice;
    string countryName, teamName, newName;

    while (true) {
        menu();
        cin >> choice;

        switch (choice) {
            case 1:
                cout << "Введіть назву країни: ";
                cin >> ws;
                getline(cin, countryName);
                add(countryName);
                break;
            case 2:
                cout << "Введіть назву країни: ";
```

```
            cin >> ws;
            getline(cin, countryName);
            cout << "Введіть назву команди: ";
            getline(cin, teamName);
            addTeam(countryName, teamName);
            break;
        case 3:
            cout << "Введіть стару назву: ";
            cin >> ws;
            getline(cin, teamName);
            cout << "Введіть нову назву: ";
            getline(cin, newName);
            renameTeam(teamName, newName);
            break;
        case 4:
            cout << "Введіть назву країни: ";
            cin >> ws;
            getline(cin, countryName);
            removeCountry(countryName);
            break;
        case 5:
            cout << "Введіть назву команди: ";
            cin >> ws;
            getline(cin, teamName);
            removeTeam(teamName);
            break;
        case 6:
            cout << "Введіть назву команди: ";
            cin >> ws;
            getline(cin, teamName);
            search(teamName);
            break;
        case 7:
            show();
            break;
        case 8:
            cout << "Вихід з програми.\n";
            clear();
            return 0;
        default:
            cout << "Невірний вибір. Спробуйте ще раз.\n";
    }
  }
}
```

**Результати программи**

```
=== Меню ===
1. Додати країну
2. Додат? команду
3. Змінити назву
4. Видалити країну
5. Видал?ти команду
6. Знайти команду
7. Показати всі дані
8. Вийти
Виберіть опцію: 7
Країна: Німеччина
  Команда: TeamD, Місце: 3
  Команда: TeamE, Місце: 6
Країна: Україна
  Команда: TeamA, Місце: 1
  Команда: TeamB, Місце: 4
  Команда: TeamC, Місце: 5
Країна: Франція
  Команда: TeamF, ?сце: 2
  Команда: TeamG, Місце: 7
  Команда: TeamH, Місце: 8
  Команда: TeamI, Місце: 9
```

```
=== Меню ===
1. Додати країну
2. До?ати команду
3. Змінити назву
4. Видалити країну
5. Ви?алити команду
6. Знайти команду
7. Показати всі дані
8. Вийти
Виберіть опцію: 6
Введіть назву команди: TeamA
Команда TeamA належ?ть країні Україна та зайняла місце 1
```

```
=== Меню ===
1. До?ати країну
2. Додати команду
3. Змінити назву
4. Вида?ити країну
5. Видалити команду
6. Знайти команду
7. П?казати всі дані
8. Вийти
Виберіть опцію: 5
Введіть назву команди: TeamB
Команду TeamB видал?но.
```

```
=== Меню ===
1. Додати країну
2. Додати команду
3. З?інити назву
4. Видалити країну
5. Видалити команду
6. Знайти команду
7. Показати всі дані
8. Вийти
Виберіть опцію: 4
Введіть назву країни: Франція
Країну Франція видалено.
```

```
=== Меню ===
1. Додати країну
2. Додати команду
3. Зміни?и назву
4. Видалити країну
5. Видалити команду
6. Зна?ти команду
7. Показати всі дані
8. Вийти
Виберіть оп?ію: 3
Введіть стару назву: TeamD
Введіть нову назву: Teamd
Команду ?TeamD не зн?йдено.
```

```
=== Меню ===
1. Додати країну
2. Додати команду
3. Змінити назву
4. Видалити країну
5. Видалити команду
6. Знайти команду
7. Показати всі дані
8. Вийти
Виберіть опцію: 2
Введіть назву країни: Україна
Введіть назву команди: 45
```

```
=== Меню ===
1. Дода?и країну
2. Додати команду
3. Змінити назву
4. Видалити країну
5. Видалити команду
6. Знайти команду
7. Показати всі дані
8. Вийти
Виберіть опцію: 1
Введіть назву країни: Польща
```

```
=== Меню ===
1. Додати країну
2. Додати команду
3. Змінити назву
4. Видалити країну
5. Видалити команду
6. Знайти команду
7. Показати всі д?ні
8. Вийти
Виберіть опцію: 7
Країна: Німеччина
  Команда: TeamD, Місце: 3
  Команда: TeamE, Місце: 6
Країна: Польща
Країна: Україна
  Команда: TeamA, Місце: 1
  Команда: TeamC, Місце: 5
  Команда: 45, Місце: не визначено
```