

Практична робота №3 (2.1)
Завірюха Еліна

```
#include <iostream>
#include <vector>
#include <algorithm>
```

```
enum class Role {
    Guardian,
    Defender,
    Midfielder,
    Attacker
};
```

```
struct PlayerData {
    std::string identity;
    Role position;
    int age;
    int matchesPlayed;
    int goalsScored;
};
```

```
void eliminatePlayer(std::vector<PlayerData>& squad, const std::string&
playerName) {
    squad.erase(std::remove_if(squad.begin(), squad.end(),
        [&playerName](const PlayerData& player) {
            return player.identity == playerName;
        }), squad.end());
}
```

```
void appendPlayerBefore(std::vector<PlayerData>& squad, const std::string&
playerName, const PlayerData& freshPlayer) {
    auto iterator = std::find_if(squad.begin(), squad.end(),
        [&playerName](const PlayerData& player) {
            return player.identity == playerName;
        });
    if (iterator != squad.end()) {
        squad.insert(iterator, freshPlayer);
    }
}
```

```
void appendPlayerAfter(std::vector<PlayerData>& squad, const std::string&
playerName, const PlayerData& freshPlayer) {
    auto iterator = std::find_if(squad.begin(), squad.end(),
        [&playerName](const PlayerData& player) {
            return player.identity == playerName;
        });
    if (iterator != squad.end()) {
```

```

        squad.insert(iterator + 1, freshPlayer);
    }
}

int main() {
    std::vector<PlayerData> squad =
    {
        {"CR7", Role::Attacker, 36, 10, 7},
        {"LM10", Role::Attacker, 34, 12, 8},
        {"Neuer", Role::Guardian, 35, 8, 0},
        {"Sergio", Role::Defender, 34, 11, 2},
        {"Luka", Role::Midfielder, 35, 7, 1},
        {"Harry", Role::Attacker, 28, 4, 3}
    };

    eliminatePlayer(squad, "Neuer"); // Removing a player named "Neuer"

    PlayerData newGuy = {"Newcomer", Role::Midfielder, 25, 3, 0};
    appendPlayerBefore(squad, "Sergio", newGuy); // Adding a new player before
    "Sergio"

    PlayerData anotherGuy = {"FreshFace", Role::Defender, 22, 2, 0};
    appendPlayerAfter(squad, "LM10", anotherGuy); // Adding a new player after
    "LM10"

    // Display the updated list of players
    std::cout << "Updated Squad:" << std::endl;
    for (const PlayerData& player : squad) {
        std::cout << "Name: " << player.identity << std::endl;
        std::cout << "Position: ";
        switch (player.position) {
            case Role::Guardian:
                std::cout << "Guardian";
                break;
            case Role::Defender:
                std::cout << "Defender";
                break;
            case Role::Midfielder:
                std::cout << "Midfielder";
                break;
            case Role::Attacker:
                std::cout << "Attacker";
                break;
        }
        std::cout << std::endl;
        std::cout << "Age: " << player.age << std::endl;
        std::cout << "Matches Played: " << player.matchesPlayed << std::endl;
        std::cout << "Goals Scored: " << player.goalsScored << std::endl;
        std::cout << "-----" << std::endl;
    }
}

```

```
    return 0;  
}
```