

# Lab 12: Usuwanie elementów

Napisz funkcję **remove**, która przyjmuje początkowy i końcowy iterator wycinka wektora liczb całkowitych oraz dowolną wartość całkowitą i kolejne elementy nierówne tej wartości kopiuje do kolejnych komórek wektora poczynając od pierwszej. Funkcja zwraca iterator końcowy wycinka powstałego ze skopiowanych elementów. Funkcja powinna być przystosowana do użycia w przykładowym programie poniżej.

Przykładowy program

```
int main() {
    std::vector<int> v {-7, 5, 2, 2, 11, 2, 3};

    auto res = remove(v.begin(), v.end(), 2);

    v.erase(res, v.end());

    for (auto it = v.cbegin(); it < v.end(); it++) {
        std::cout << *it++ << " ";
    }

    std::cout << std::endl;

    return 0;
}
```

Wykonanie

Out: -7 5 11 3

# Lab 12: Sortowanie wycinka wektora

Sortowanie wektora w kolejności niemalejącej przez wybór przebiega następująco. Znajdujemy w wektorze element najmniejszy i zamieniamy go miejscami z pierwszym. Następnie powtarzamy te czynności dla wektora bez pierwszego elementu i tak dalej. Napisz funkcję **selection\_sort**, która przyjmuje początkowy oraz końcowy iterator wycinka wektora liczb całkowitych i sortuje ten wycinek niemalejąco przez wybór. Funkcja powinna być przystosowana do użycia w przykładowym programie poniżej.

Przykładowy program

```
int main() {
    std::vector v {13, -2, 21, 5, -8, 5, 7, -10};

    selection_sort(v.begin(), v.end());

    for (auto it = v.cbegin(); it < v.cend(); ) {
        std::cout << *it++ << " ";
    }

    std::cout << std::endl;

    return 0;
}
```

Wykonanie

Out: -10 -8 -2 5 5 7 13 21