

Lab 13: Sortowanie wycinka wektora w dowolnej kolejności

Napisz funkcję **selection_sort**, która przyjmuje początkowy i końcowy iterator wycinka wektora liczb całkowitych oraz funktor porównujący dwie takie liczby i sortuje ten wycinek przez wybór, w kolejności zadanej funktorem porównującym. Przy braku trzeciego argumentu sortowanie odbywa się w kolejności niemalejącej. Korzystając z tej funkcji napisz program sortujący jeden wycinek wektora niemalejąco, a inny nierosnąco. Porządek sortowania zadaj wyrażeniami lambda.

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

    selection_sort (vec.begin(), vec.begin()+4);

    selection_sort (vec.begin()+4, vec.end(),
        [](int a, int b) {return a > b; }
    );

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

    std::cout << std::endl;

    return 0;
}
```

Lab 13: Powtarzanie dla każdego elementu

Napisz funkcję **for_each**, która przyjmuje początkowy i końcowy iterator wycinka wektora liczb całkowitych oraz funktor przyjmujący referencję takiej liczby i wywołuje ten funktor na wszystkich elementach wycinka. Korzystając z tej funkcji napisz program, który tworzy pięcioelementowy wektor liczb całkowitych, wczytuje jego elementy ze standardowego wejścia, i wypisuje na standardowe wyjście wraz z indeksami. Jako trzeci argument funkcji **for_each** przekaż wyrażenie lambda.

Przykładowe wykonanie

In: 7 3 9 2 6

Out: 0 7

Out: 1 3

Out: 2 9

Out: 3 2

Out: 4 6