

Container, Iteratoren und Algorithmen

☑ **Aufgabe 10.1** Versuchen Sie, das folgende Programm in Ihrem Kopf auszuführen, ohne dabei einen Compiler zu Hilfe zu ziehen. Was ist der ausgegebene Inhalt des Vektors `v` in den einzelnen Programmzeilen der `main`-Funktion?

```
void print(const std::vector<int>& v) {
    for(int x : v)
        std::cout << x << " ";
    std::cout << std::endl;
}

int main() {
    std::vector<int> v = {1,2,3,4,5,6,7,8,9};           print(v);
    std::reverse(v.begin(), v.end());                  print(v);
    std::replace(v.begin(), v.end(), 8, 3);             print(v);
    std::fill(v.begin()+3, v.end()-4, 3);              print(v);
    std::copy(v.begin()+1, v.begin()+4, v.end()-3);    print(v);
    std::sort(v.rbegin(), v.rend());                   print(v);
    std::remove(v.begin(), v.end(), 7);                print(v);
}
```

☑ **Aufgabe 10.2** Jede einzelne von fünf Programmzeilen in der `main`-Funktion von folgendem Programm führt zu einem Compiler-Fehler. Erklären Sie jeweils, wie dieser Fehler zustande kommt und wie man das Programm anpassen könnte, so dass es sich wie gewünscht verhält.

```
struct Person {
    std::string m_name;
};

int main() {
    std::list<int> l = {3,2,1}; std::sort(l.begin(), l.end());
    std::vector<int> v = {3,2,1}; std::fill(v.cbegin(), v.cend(), 0);
    std::forward_list<int> f = {3,2,1}; std::reverse(f.begin(), f.end());
    std::set<Person> s = {{ "Tom" }, { "Amy" }}; s.insert({ "Bob" });
    std::unordered_set<Person> u = {{ "Eva" }, { "Noa" }}; u.erase({ "Noa" });
}
```

⚠ **Aufgabe 10.3** Implementieren Sie eine Funktion `update_max_element`, die in einem gegebenen Behälter von Werten zuerst das Maximum findet und dann diesen maximalen Eintrag mit einem neuen Wert überschreibt. Falls gleich mehrere Einträge denselben maximalen Wert haben, dann sollen alle mit dem neuen Wert überschrieben werden.

Etwas genauer formuliert soll diese Funktion die folgenden drei Argumente haben:

1. Einen Iterator, der auf das erste Element des Behälters zeigt.
2. Einen Iterator, der auf die Position gerade nach dem letzten Element des Behälters zeigt.
3. Einen Wert, mit dem das gefundene Maximum überschrieben werden soll.

Überlegen Sie sich selbstständig, wie man die gewünschte Funktion möglichst generisch als Funktionen-Template umsetzen kann und wie viele Typ-Parameter dazu gebraucht werden. Denken Sie ebenso darüber nach, was die genauen Anforderungen sind an den Typ des Iteratoren-Paars und an den Typ der enthaltenen Werte. Geben Sie insbesondere an, welche Iteratoren-Kategorie für die Implementation dieser Funktion benötigt wird.