

# Aufgabensammlung zu STL Algorithmen und Containern

Universität Wien, Fakultät für Informatik

Nachfolgend finden Sie eine Aufgabensammlung um [STL Algorithmen](#)<sup>1</sup> sowie den Umgang mit Containern wie, z.B. `set` und `unordered_map` zu üben. Beachten Sie, dass es noch weitere Arten von [Containern](#) gibt<sup>2</sup> und Sie die nachfolgenden Aufgaben ebenfalls für diese anpassen können.

Als weitere Übung, können Sie auch die Objekte in ihre Vector-Klasse einlesen, statt den STL-Vector zu verwenden. Die entsprechenden Files (`miniatur.o`, `minitaure.h`, `main.cpp`, `gen`) finden Sie unter

`/home/Xchange/PR2/ue4/`

Beachten Sie, dass das `.o`-File in der Regel nur auf dem almighty getestet werden kann, da es umgebungsspezifisch ist. Eine Anleitung zum Verbinden auf den almighty steht auf Moodle zur Verfügung.

Kopieren Sie die Dateien in Ihr home-Directory am almighty. Um ein `objects.txt` File zu generieren führen Sie `gen` aus. Hierbei ist zusätzlich die zu generierende Anzahl an Objekten anzugeben, z.B., `.\gen 10`.

Versuchen Sie zu Übungszwecken kein copy-paste zwischen den Aufgaben zu verwenden, Schleifen sowie `for_each` zu vermeiden und passende Algorithmen zu verwenden.

1. Speichern Sie die Elemente aus `Vector<Miniatur> v` in einem `set<string>` Objekt das lexikografisch absteigend sortiert ist. Die Einträge im Set sollen wie folgt aussehen: "Tyraniden Genestrealer 10", etc..
2. Speichern Sie die Elemente aus `Vector<Miniatur> v` in einem `set<string>` Objekt das nach der Länge der Strings absteigend sortiert ist. Die Einträge im Set sollen wie folgt aussehen: "Tyraniden Genestrealer 10", etc..
3. Wiederholen Sie Aufgabe 1 und 2 mit aufsteigender Sortierung.
4. Denken Sie sich weitere Sortiermöglichkeiten aus, z.B. auch mehrstufige Verfahren.
5. Ermitteln Sie die Summe aller Punkte von Miniaturen aus Vektor `v`, welche ein bestimmtes Zeichen (`char`) im Namen haben. (Das Zeichen soll vorher mittels `cin` eingelesen werden.)
6. Erstellen Sie eine `unordered map<Fraktion, int>`, in der Sie alle Werte aus Vektor `v` speichern. Der Schlüssel ist dabei die Fraktion einer Miniatur. Der Wert soll die Gesamtsumme aus *Länge des Namens + Punkte für alle zugehörigen Miniaturen* sein. `for_each` ist erlaubt.
7. Ändern Sie alle Strings aus einem der vorherigen Sets (`set<string>`) in Strings aus nur Großbuchstaben und kopieren Sie diese in einen Vektor.
8. Ändern Sie alle Strings aus einem der vorherigen Sets (`set<string>`) in Strings aus nur Kleinbuchstaben und kopieren Sie diese in einen Vektor.
9. Bestimmen Sie den längsten String aus einem der vorherigen Sets.
10. Bestimmen Sie den kürzesten String aus einem der vorherigen Sets.

---

<sup>1</sup>siehe auch PR2 VO Folien S.30-39

<sup>2</sup>siehe auch PR2 VO Folien S.43-53

11. Bestimmen Sie aus einem der vorherigen Sets den längsten String zwischen dem ersten String welcher ein bestimmtes Zeichen (**char**) beinhaltet (das Zeichen soll vorher mittels **cin** eingelesen werden) und dem letzten Element des Sets.
12. Bestimmen Sie aus einem der vorherigen Sets den kürzesten String zwischen dem ersten String welcher ein bestimmtes Zeichen (**char**) beinhaltet (das Zeichen soll vorher mittels **cin** eingelesen werden) und dem letzten Element des Sets.
13. Bestimmen Sie aus **Vector<Miniatur> v** die Menge der Elemente die der Fraktion **Tyraniden** angehören und mindestens einmal ein bestimmtes Zeichen (**char**) im Namen haben (das Zeichen soll vorher mittels **cin** eingelesen werden).
14. Bestimmen Sie aus **Vector<Miniatur> v** die Menge der Elemente die der Fraktion **Tyraniden** angehören und ein bestimmtes Zeichen (**char**) nicht im Namen haben (das Zeichen soll vorher mittels **cin** eingelesen werden).