



## Einführung in die Programmierung für Studierende der Naturwissenschaften

Blatt 9 – 04.07.2022

Abgabe: bis Montag, 11. Juli, 16 Uhr per e-Mail an Ihre:n Tutor:in.

---

**Aufgabe 1** (5 Punkte). Schreiben Sie ein Programm, in dem Sie am Anfang z.B. mit

```
1 using Container = std::list<int>;
```

einen Container-Typ festlegen. Füllen Sie zur illustration der Funktion Ihres Programms einen Container mit den Zahlen von 1 bis 10. Verwenden Sie anschließend Iteratoren vom Typ `Container::iterator` um über die Liste zu iterieren und alle Werte durch ihr Quadrat zu ersetzen.

**Aufgabe 2.** Schreiben Sie eine Funktion

```
1 template <class Iterator> Iterator  
2     find_min(Iterator begin, Iterator end) {  
3     // TODO implement  
4 }
```

die die Position des kleinsten Elements in einem Iterator-Bereich findet.

*Hinweis:* Sie können die Datei `find_min.cc` von der Vorlesungswebseite herunterladen und dort die Funktion "ausfüllen". Der Test in der `main()`-Funktion sollte dann unabhängig davon funktionieren, ob Sie anstatt mit einer `std::list` z.B. mit einem `std::vector` arbeiten.

**Aufgabe 3** (5 Punkte). Mit Hilfe einer `std::priority_queue` kann eine Liste der Länge  $N$  sortiert werden, in dem erst alle Elemente nacheinander mit `push` in die queue eingefügt werden und dann mit `top` einzeln ausgelesen und mit `pop` wieder aus der `std::priority_queue` gelöscht werden.

- (a) Erläutern Sie, warum das Verfahren die Daten sortiert
- (b) Implementieren Sie das Verfahren unter Verwendung der `std::priority_queue`. Es ist ausreichend, wenn Sie das Verfahren für Daten vom Typ `int` implementieren. Die Daten dürfen auch in absteigender Reihenfolge sortiert werden. Die Daten müssen nicht sortiert wieder gespeichert werden, es reicht, wenn sie sortiert ausgegeben werden.

*Hinweis:* Es ist Teil der Aufgabe, dass Sie sich mit der Funktionsweise von `std::priority_queue` vertraut machen. Eine gute Anlaufstelle ist die Referenz unter [www.cppreference.com](http://www.cppreference.com).

**Aufgabe 4** (5 Punkte). Verwenden Sie OpenMP um eine Funktion

```
1     int parallel_min(int* L, unsigned int n, unsigned int np)
2     {
3         // ...
4     }
```

zu schreiben, die `np` Threads benutzt, um das Minimum in einer Liste `L` der Länge `n` zu finden. *Hinweis:* hier sollen keine Iteratoren verwendet werden.

**Aufgabe 5** (optional). Schildern Sie in einer Datei `erfahrung.txt` kurz Ihre Erfahrung mit dem aktuellen Übungsblatt. Berichten Sie darin wieder in Stichpunkten bzw. ein bis zwei kurzen Sätzen über Ihre Erfahrungen mit Kursinhalt und Übungsaufgaben. Was fiel Ihnen leicht? Was ist noch unklar? Wie viel Zeit haben Sie für die Bearbeitung der Hausaufgaben benötigt und welche Probleme traten dabei auf?