



Programmierpraktikum Technische Informatik (C++)

Aufgabe 09

Hinweise

Abgabe: Stand des Git-Repositories am 7.7.2020 um 9 Uhr.

Die Dateien zur Bearbeitung dieser Aufgabe erhalten Sie, indem Sie die neue Aufgabe aus dem Aufgabenrepository in Ihr lokales mergen. Dies geschieht mit `git pull cpp2020 master` innerhalb Ihres Repositories. Die Lösungen committen Sie bitte in Ihr lokales Repository (`git commit -a` oder `git add` gefolgt von `git commit`) und pushen sie in Ihr Repository auf dem git-Server des Instituts (`git push`).

Teilaufgabe 1 (2 Punkte)

Die folgende Aufgabe behandelt das Logik-Spiel der "Türme von Hanoi" (https://en.wikipedia.org/wiki/Tower_of_Hanoi). Die zu implementierenden Methoden finden Sie in der Datei `hanoi.cpp`.

- a) Implementieren Sie die Methode `moveTopPlate`. Diese soll die oberste Platte des ersten übergebenen Turmes auf den zweiten verschieben. Stellen Sie sicher, dass der Zug legal ist, sprich keine größere Platte auf eine kleinere verschoben wird. Vor dem Verschieben soll die übergebene Funktion `preMovePrint` und nach dem Zug `postMovePrint` ausgeführt werden.
- Hinweise:
- Die übergebenen Parameter der `preMovePrint`-Funktion können die Namen der beiden übergebenen Türme sein (mehr dazu in späteren Teilaufgaben).
- a) Implementieren Sie die Methode `solveHanoi`. Diese wird rekursiv mit immer kleiner werdenden Turmgrößen aufgerufen bis das minimale Problem eines Turmes der Größe 1 einfach gelöst werden kann. Allerdings müssen auch der Ziel-, Aushilfs- und Ausgangsturm angepasst werden:
- Ist die Turmgröße 1, so wird die oberste Platte vom Ausgangs- auf den Zielturm verschoben.
 - Sonst wird die Funktion rekursiv mit einer um 1 verringerten Turmgröße aufgerufen. Der Ziel- und Aushilfsturm werden dabei getauscht.



- Danach wird die Platte vom Ausgangs- zum Zielturm verschoben.
 - Zuletzt wird die Funktion noch einmal rekursiv mit einer um 1 verringerten Turmgröße aufgerufen, dieses mal mit dem Aushilfs- und dem Ausgangsturm vertauscht.
- a) Implementieren Sie die Methode `solveHanoiStart`. Diese soll zwei Lambda-Funktionen definieren, die in der Rekursion benötigt werden, und dann die Rekursion starten. In der Funktion `preMovePrint` soll auf dem übergebenen Outputstream der geplante Zug ausgegeben werden. Sie soll dabei ausgeben, von welchem auf welchen Turm verschoben wird. In der `postMovePrint`-Funktion sollen die Inhalte der Türme übersichtlich ausgegeben werden. Insgesamt ist Ihnen bei beiden Funktionen sehr viel kreativer Freiraum gelassen und Sie können, falls nötig, die Signaturen der beiden Funktionen anpassen.

Teilaufgabe 2 (3 Punkte)

In der folgenden Aufgabe soll ein Programm erstellt werden, das Primzahlen findet. Der hierfür genutzte Algorithmus ist das Sieb des Eratosthenes

(https://en.wikipedia.org/wiki/Sieve_of_Eratosthenes). Die zu implementierenden Methoden finden Sie in der Datei `sieve.cpp`.

- a) Implementieren Sie die Methode `reduceToPrimes`. Diese soll den `numbers`-Vektor mithilfe des SdE-Algorithmus auf alle darin enthaltenen Primzahlen reduzieren. Der Algorithmus läuft wie folgt ab: Für jede Zahl aufsteigend mit 2 beginnend lösche alle Vielfachen der Zahl aus dem Vektor heraus.

Hinweise:

- Der `numbers`-Vektor wurde mit den Zahlen zwischen 2 und 100 initialisiert, ist jedoch unsortiert.
 - Die Standardbibliothek bietet viele verschiedene Algorithmen an. Benutzen Sie einen geeigneten Algorithmus, um den `numbers`-Vektor zu sortieren.
 - Die Vielfachen der Zahl können durch die Vektor-Methode `erase` und den Standard-Algorithmus `remove_if` entfernt werden.
- a) Implementieren Sie die Methode `resetToSize`. Sie soll zunächst alle Zahlen, die größer als der übergebene Wert sind, aus dem `numbers`-Vektor löschen. Danach sollen ihm alle Nummern zwischen 2 und der übergebenen Größe hinzugefügt werden, die er nicht bereits enthält. Nach dem Ablauf dieser Methode sollen die Zahlen im `numbers`-Vektor außerdem in absteigender Reihenfolge vorliegen.
- Hinweise:
- Der `numbers`-Vektor soll nicht durch einen neuen Vektor ersetzt, oder komplett



geleert und dann neu befüllt werden.

- Sortierreihenfolgen von Sortieralgorithmen der Standardbibliothek können durch Lambdafunktionen beeinflusst werden.

- a) Implementieren Sie die Methode `groupNumbers`. Diese soll alle Zahlen zwischen den übergebenen Grenzen (die bereits im Vektor vorhanden sind) an den Beginn des `numbers`-Vektors bewegen. Die aufsteigende Reihenfolge soll dabei beibehalten werden.

Hinweis:

- Diese Methode kann mit genau einem Aufruf eines Standard-Algorithmus gelöst werden. Es bietet sich hierzu der Algorithmus `std::stable_partition` an.