

Übung 6

Ziel der Übung:

- Wiederholung: Programmierung, insbesondere Funktionen
- Call-by-reference Funktionsaufruf
- Rekursive Funktionen

Aufgabe 1: Funktion (Call-by-reference) zur Berechnung mehrerer Werte

In der letzten Übung haben Sie zwei Funktionen geschrieben, die den Oberflächeninhalt und das Volumen einer Kugel berechnen. Es sollen jetzt beide Berechnungen **in einer Funktion** implementiert werden. Schreiben Sie eine C++-Funktion, die als Eingabeparameter den Radius einer Kugel erhält und den Oberflächeninhalt und das Volumen berechnet. In der aufrufenden main-Funktion sollen die Ergebnisse beider Berechnungen ausgegeben werden. Sie können das Hauptprogramm aus der letzten Übung verwenden und den Funktionsaufruf anpassen, um diese Funktion aufzurufen und zu testen.

Aufgabe 2: Rekursive Funktion für die Berechnung der Collatz-Folge

Die Collatz-Folge beginnt mit einer beliebigen natürlichen Zahl $n \geq 1$. Ist n gerade, so erhält man das nächste Folgeglied durch Halbierung von n . Ist n ungerade, so erhält man das nächste Folgeglied dadurch, dass man n verdreifacht und um eins erhöht. Die Folge endet, wenn ein Glied den Wert 1 hat. D.h. solange das Folgeglied $n_i \neq 1$ ist, lässt sich das nächste Glied n_{i+1} der Collatz-Folge rekursiv wie folgt berechnen:

$$\text{collatz}(n_{i+1}) = \begin{cases} \text{collatz}\left(\frac{n_i}{2}\right) & \text{falls } n_i \text{ gerade ist} \\ \text{collatz}(3 * n_i + 1) & \text{falls } n_i \text{ ungerade ist} \end{cases}$$

Teil a) Schreiben Sie eine C++-Funktion `void collatz(int n)`, welche die Glieder der Collatz-Folge einer natürlichen Zahl n nach der oben beschriebenen Vorschrift rekursiv berechnet und ausgibt, d.h. die Ausgabe eines Folgegliedes erfolgt in dem zugehörigem Rekursionsschritt, so dass nach der Terminierung der Funktion die gesamte Folge auf dem Bildschirm angezeigt wird.

Teil b) Schreiben Sie ein C++-Programm, das eine natürlichen Zahl $n \geq 1$ einliest und als Startwert an die Funktion `collatz` übergibt.

Beispiele:

Geben Sie eine Zahl ein: 7

7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

Geben Sie eine Zahl ein: 30

30 15 46 23 70 35 106 53 160 80 40 20 10 5 16 8 4 2 1

Teil c) Geben Sie mit Ihrer rekursiven Funktion die Folge in umgekehrter Reihenfolge aus.

Hinweis: Es stellt sich die Frage, ob die Collatz-Folge für alle natürlichen Zahlen als Startwert endlich ist. Man vermutet, dass dies tatsächlich so ist, da bis heute kein Startwert bekannt ist, für den die Folge nicht endlich ist.

Hausaufgaben Serie 6

Hausaufgaben mit Namen, Studiengang und Matrikelnummer unter „**Aufgaben**“ auf StudIP hochladen.
Abgabe bis **30.11.2021** (ÜG-1), **01.12.2021** (ÜG-4), **02.12.2021** (ÜG-2) und **03.12.2021** (ÜG-3).

Aufgabe 1: Sortieren mit Austauschen

Ein einfacher Algorithmus zum Sortieren von Elementen besteht darin, benachbarte Elemente zu vergleichen und zu vertauschen.

Teil a) Schreiben Sie eine C++-Funktion **tauschen**, die zwei Ganzzahlen übergeben bekommt und miteinander vertauscht.

Teil b) Schreiben Sie eine C++-Funktion **sortieren**, die drei Ganzzahlen a , b und c übergeben bekommt und mit Hilfe von `if`-Anweisungen der Reihe nach aufsteigend sortiert. Für das Austauschen zweier Ganzzahlwerte soll die Funktion **tauschen** verwendet werden. Sind alle Zahlen sortiert, soll die Funktion die Differenz zwischen der größten und der kleinsten Zahl als Ergebnis berechnen und zurückgeben.

Teil c) Schreiben Sie ein C++-Programm, das drei Ganzzahlen einliest und an die Funktion **sortieren** übergibt. Danach sollen die drei Ganzzahlen in sortierter Reihenfolge sowie die Differenz ausgegeben werden.

Für die Lösung dieser Aufgabe sollen noch **keine Arrays** verwendet werden.

Tip: Nutzen Sie die Verwendung von Referenzparametern („call-by-reference“) für die Übergabe der Parameter an die beiden Funktionen.

Beispiel:

Geben Sie Zahl a ein: 14
Geben Sie Zahl b ein: -5
Geben Sie Zahl c ein: 2
Sortiert: -5, 2, 14
Differenz: 19

(12 Punkte)

Aufgabe 2: Rekursive Berechnung der Quadratwurzel einer Zahl

Nach dem Heron-Verfahren lässt sich die Näherung der Quadratwurzel einer Zahl a mit der folgenden Rekursionsvorschrift berechnen, wobei n die Rekursionstiefe angibt:

$$w(n, a) = \begin{cases} \frac{1}{2} \left[w(n-1, a) + \frac{a}{w(n-1, a)} \right] & \text{falls } n \geq 1 \\ 1 & \text{falls } n = 0 \end{cases}$$

Teil a) Schreiben Sie eine rekursive C++-Funktion **double wurzel(int n, double a)** zur Berechnung der Quadratwurzel nach dem oben beschriebenen Heron-Verfahren.

Teil b) Schreiben Sie ein C++-Programm, das eine reelle Zahl a und die Rekursionstiefe n einliest. Wenn a und n größer gleich 1 sind, sollen beide als Parameter an die Funktion **wurzel** übergeben werden, andernfalls wird eine Fehlermeldung ausgegeben. Das Ergebnis und der exakte Wert der Wurzelberechnung sollen ausgegeben werden.

Beispiele:

Geben Sie eine Zahl ein: 20

Geben Sie die Anzahl der Rekursionen ein: 3

wurzel(3, 20): 4.71347

sqrt(20): 4.47214

Geben Sie eine Zahl ein: 25

Geben Sie die Anzahl der Rekursionen ein: 0

Fehler

(8 Punkte)