
Übung zur Vorlesung
Informatik I (B.Sc. Physik)
SoSe 2020

Aufgabenblatt 3: Funktionen und Rekursion

Ausgabe: 16.06.2020

Abgabe: 23.06.2020

Die Besprechung der Übungsaufgaben findet eine Woche nach Abgabetermin als Video-Konferenz im Übungszeitraum statt. Beachten Sie die Hinweise zur Übung aus der Einführungsveranstaltung!

Aufgabe 1 Berechnung von π

(6 Punkte)

Nach Leibniz kann die Zahl π durch folgende Reihe mathematisch beschrieben werden:

$$\pi = 4 \cdot \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} = 4 \cdot \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots\right)$$

Schreiben Sie eine Funktion `double calcPi(double precision)`, die π auf eine gegebene Genauigkeit nach diesem Verfahren berechnet! Der Funktion soll ein Wert `precision` für die Genauigkeit übergeben werden. Die Iteration bricht ab, wenn sich das Ergebnis in einem Schritt um weniger als `precision` ändert. Vergleichen Sie (im Hauptprogramm) den berechneten Wert mit `M_PI` (`#include <cmath>`) und geben den Fehler aus.

Hinweis: Die Formel $\frac{(-1)^n}{2n+1}$ ist mathematisch korrekt, bei direkter Umsetzung aber rechenaufwendig. Eine effizientere Implementierung könnte den Nenner und das Vorzeichen in jedem Schritt anpassen.

Aufgabe 2 Erzeugen der binären Zahlendarstellung

(7 Punkte)

Setzen Sie in der Programmiersprache C++ eine Funktion um, die zur Umwandlung von Dezimalzahlen in die binäre Zahlendarstellung genutzt werden kann. Die Funktion bekommt als Parameter eine ganze Zahl $n \geq 0$ und liefert die entsprechende binäre Darstellung von n als Ergebnis. Diese binäre Darstellung soll ebenfalls in einem ganzzahligen Datentyp zurückgegeben werden.

Hinweis: Die Binärzahlendarstellung lässt sich durch eine Variable eines ganzzahligen Datentyps repräsentieren, indem Sie sicherstellen, dass nur die Ziffern 0 und 1 verwendet werden.

Der Nutzer soll im Hauptprogramm eine ganze Zahl $n \geq 0$ in Dezimalzahlendarstellung eingeben können, für die das Programm die zugehörige binäre Darstellung über obige Funktion berechnet und anschließend auf der Konsole ausgibt.

Ermöglichen Sie eine wiederholte Ausführung / weitere Umwandlungen, solange die Eingabe nichtnegativ ist (Programm bei Eingabe einer negativen Zahl beenden).

Aufgabe 3 Quersumme und Querprodukt

(5 Punkte)

Die Quersumme einer ganzen Zahl entspricht der Summe ihrer Ziffernwerte. Analog lässt sich auch das Querprodukt als Produkt der Ziffernwerte definieren. So ist beispielsweise 13 die Quersumme und 90 das Querprodukt der ganzen Zahl 2353.

- Schreiben Sie eine Funktion `quer`, die für eine positive ganze Zahl sowohl die Quersumme als auch das Querprodukt berechnet.
- Verwenden Sie diese Funktion im Hauptprogramm, welches alle Zahlen zwischen 1 und 100 auf den Bildschirm ausgeben soll, für die gilt, dass die Summe von Quersumme und Querprodukt wieder die Zahl selbst ergibt (beispielsweise 39, da $(3 \cdot 9) + (3 + 9) = 39$).

Aufgabe 4 Binomialkoeffizienten

(7 Punkte)

Eine anschauliche Methode zur Bestimmung der Binomialkoeffizienten $\binom{m}{n}$ ist das Pascalsche Dreieck.

| | | | | | | |
|---|---|---|---|---|---|-----|
| | | | 1 | | | m=0 |
| | | 1 | | 1 | | m=1 |
| | 1 | | 2 | | 1 | m=2 |
| 1 | | 3 | | 3 | | m=3 |
| 1 | 4 | | 6 | | 4 | m=4 |
| . | . | . | . | . | . | . |

Dabei ergeben sich die Elemente der nächsten Zeile als Summe der zwei darüberstehenden Elemente.

Formulieren Sie diese rekursive Berechnung eines Binomialkoeffizienten als C++-Funktion!

Benutzen Sie diese Funktion in einem Hauptprogramm, welches das Pascalsche Dreieck (mit mindestens 5 Zeilen) ausgibt!

Versuchen Sie durch geschicktes Ausgeben von Leerzeichen eine optisch ansprechende Darstellung der Ausgabe zu erzeugen.

Viel Erfolg!