

Programmieren für Physiker

Interfakultatives Institut für Anwendungen der Informatik
Institut für Theoretische Teilchenphysik

Prof. Dr. M. Steinhauser, Dr. A. Mildenerger
<http://comp.physik.kit.edu>

SS 2011 – Blatt 08
Bearbeitungszeitraum: bis 08. Juni MMXI

Aufgabe XIX: Römisch → Arabisch

Pflichtaufgabe

Klasse, endlich hat es mit der Rom-Reise geklappt. Aber nun fragt Sie Ihre altphilologische Reisebegleitung ständig, ob Sie mal eben die römische Jahreszahl an der oder jener Inschrift umrechnen könnten. Ach, was wäre es toll, ein Programm dafür zu haben...

Na denn. Es gibt sieben römische Zahlzeichen:

römisch	M	D	C	L	X	V	I
Wert	1000	500	100	50	10	5	1

Im Wert absteigende oder gleich große Zahlzeichen werden einfach addiert (Bsp. VII \equiv 7). Falls aber ein kleinerer Zahlwert vor einer größeren Zahl auftritt, wird die kleinere Zahl subtrahiert (Bsp. IV \equiv 4). Zwei direkt aufeinanderfolgende Subtraktionen kommen in regulär gebildeten Zahlen nicht vor.

Lassen Sie den Benutzer ein `char`-Feld eingeben. Nach `#include <cstring>` können Sie mit `strlen` die Länge des Zahlentexts ermitteln. Rechnen Sie dann mit oben angegebener Regel (Addition oder Subtraktion der einzelnen Zeichenwerte) die Gesamtzahl aus und geben Sie diese aus. Sie brauchen nicht zu untersuchen, ob die eingegebene römische Zahl einen gültigen regulären Aufbau besitzt.

Schreiben und verwenden Sie bitte folgende Funktionen: Eine Funktion `char to_upper (char)`, die ein Buchstaben-Argument als Großbuchstaben zurück gibt. Weiterhin eine Funktion `int value_of_rmchar (char)`, die von einem einzelnen römischen Zahlzeichen den Wert nach obiger Tabelle ermittelt. Greifen Sie auf die Funktion `to_upper` zurück, so dass Ihr Programm für Groß- und Kleinbuchstaben funktioniert.

Zusatz (freiwillig): Die umgekehrte Fragestellung: Wie kann man denn eine eingegebene arabische Zahl in die römische Zahldarstellung umrechnen?

Aufgabe 20: Minuten-Sekunden-Struct

Pflichtaufgabe

Schreiben Sie mittels `struct` einen Datentyp `minsek`, der zur Aufbewahrung von Zeiten mit Minuten- und Sekundenwert dient und dazu zwei `int`-Variablen verwendet.

Programmieren Sie bitte folgende drei Funktionen, die jeweils `minsek` Argumente oder Rückgabewerte verwenden: (a) Eingeben einer Zeit, dabei sollen Minuten und Sekunde eingegeben werden. (b) Die Ausgabe in Form Minuten:Sekunden eines derartigen Structs. (c) Die Addition von zwei Zeiten, das Ergebnis soll wieder vom Typ `minsek` sein. Berücksichtigen Sie dabei bitte eventuelle Überträge der Sekundenvariable zu den Minuten, so dass die Sekundenvariable stets im Intervall $[0, 59]$ ist.

Rufen Sie im Hauptprogramm diese Funktionen so auf, dass Ihr Programm zwei eingegebene Zeiten addiert.

Aufgabe 21: Quersumme

freiwillig

Schreiben Sie eine rekursive Funktion, die die Quersumme, also die Summe der Ziffern, einer positiven Zahl berechnet.

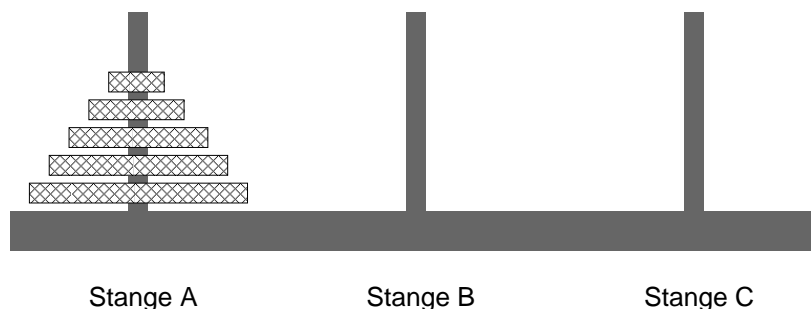
Verwenden Sie diese Funktion, um eine zweite rekursive Funktion zu erstellen, die die vollständig reduzierte Quersumme berechnet. Letztere entsteht durch wiederholtes Bilden der Quersumme, bis das Ergebnis einstellig ist.

Lassen Sie den Benutzer im Hauptprogramm eine Zahl eingeben und geben Sie die Quersumme und die vollständig reduzierte Quersumme aus.

Aufgabe 22: Türme von Hanoi

freiwillig

Das Spiel *Türme von Hanoi* besteht aus n verschiedenen großen Scheiben und drei Stangen. Zu Beginn liegen alle Scheiben, der Größe nach sortiert, auf der ersten Stange. Ziel des Spieles ist es, alle Scheiben von der ersten auf eine andere Stange zu versetzen. Dabei sind jedoch folgende Regeln zu beachten: Es darf immer nur eine Scheibe zu einem Zeitpunkt bewegt werden und es darf nie eine größere auf eine kleinere Scheibe gelegt werden. Eine Skizze der Ausgangsposition mit 5 Scheiben:



Um ein Gefühl für die Problemstellung zu erhalten, ist es ratsam, sich zunächst die Lösung für $n = 1, 2, 3$ und evtl. $n = 4$ zu überlegen.

Ist Ihnen aufgefallen, dass Sie das Vorgehen für n Scheiben auf die Lösung für $n - 1$ Scheiben zurückführen können? Nehmen Sie an, Sie hätten ein Verfahren, um die obersten $n - 1$ Scheiben zu versetzen. Welche Schritte sind dann nötig, um den Turm der Größe n zu versetzen? Dieser Lösungsweg kann mittels Rekursion in einem Programm umgesetzt werden.

Entwickeln Sie ein Programm, das für ein einzugebendes n alle einzelnen Scheibenbewegungen ausgibt, die nötig sind, um das Problem zu lösen.
