

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 2012 – Blatt 08
Bearbeitungszeitraum: bis 13. Juni MMXII

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: Nullstellenbestimmung

freiwillig

Eine Methoden, um Nullstellen einer (stetigen) Funktion numerisch zu bestimmen, ist das Bisektionsverfahren. Bei dem Verfahren startet man mit einem Intervall $[x_0, x_1]$, das eine Nullstelle der gegebenen Funktion $f(x)$ einschließen soll. Lassen Sie also die Werte x_0 und x_1 eingeben und überprüfen Sie anhand der Vorzeichen von $f(x_0)$ und $f(x_1)$, ob eine Nullstelle im Intervall vorhanden ist.

Als neue Teststützstelle x_{test} wird nun die Mitte des Intervalls herangezogen $x_{\text{test}} = (x_0 + x_1)/2$. An der neu gewonnen Stelle ist die Funktion auszuwerten und abhängig vom Vorzeichen wird entweder die obere oder die untere Intervallgrenze neu gesetzt, so dass die Nullstelle stets innerhalb des betrachteten Intervalls bleibt. Wiederholen Sie diese Schritte, bis entweder die Intervallgröße kleiner als $\varepsilon_x = 10^{-4}$ oder der Funktionswert $|f(x_{\text{test}})| < \varepsilon_f = 10^{-3}$ wird.

Ihr Programm soll folgendermaßen strukturiert sein: Definieren Sie für die zu untersuchende Funktion $f(x)$ eine C++-Funktion und schreiben Sie für die Nullstellensuche eine Funktion, die als Übergabewerte das Anfangsintervall und die gewünschten Genauigkeitswerte erhält. Geben Sie während der Iteration bei jedem Schritt die Schrittzahl, das Nullstellen-Intervall und den Funktionswert $f(x_{\text{test}})$ aus.

Testfunktionen: (i) $f(x) = \cos(x) - x$, $x_0 = -10$, $x_1 = 10$.

(ii) $f(x) = e^x - x^3$, $x_0 = 2$, $x_1 = 10$ (double-Arithmetik empfohlen).

Aufgabe 22: 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.

Hinweis:

Auf den Webseiten befindet sich ein Informationsblatt, auf welchem die Namen und Bedeutungen von wichtigen mathematischen Funktionen in C++ aufgelistet sind.
