Dirk Praetorius
Christoph Überhuber

Wintersemester 2006/07
20. Oktober 2006

Übungen zur Vorlesung
Einführung in das Programmieren für TM

Serie 2

Die Aufgaben mit Stern (*) sind bis zur nächsten Übung vorzubereiten und werden dort abgeprüft. Die übrigen Aufgaben dienen nur Ihrer Übung und mir als zusätzliche Grundlage für den Prüfungsstoff in den schriftlichen Tests. Kopieren Sie bitte den Source-Code auf Ihren Account auf der `cad.zserv.tuwien.ac.at` *in ein Unterverrzeichnis* `serie02` *Ihres Home-Verzeichnisses. Überprüfen Sie bitte vor der Übung, ob Ihre Source-Codes mit dem* `gcc` *compiliert werden können. In den folgenden Aufgaben 9, 15 und 16 sollen noch einmal* ***Verzweigungen*** *geübt werden. Die Aufgaben 10 - 14 beschäftigen sich mit* ***rekursiven Funktionen****.*

Aufgabe 9*. Schreiben Sie eine Prozedur `quadrant`, die für einen Punkt $(x, y) \in \mathbb{R}^2$ ausgibt, ob (x, y) auf einer der Achsen des Koordinatensystems liegt. Falls nicht, soll ausgegeben werden, in welchem Quadranten (x, y) liegt. Schreiben Sie ferner ein Hauptprogramm, in dem $x, y \in \mathbb{R}$ eingelesen werden. Speichern Sie den Source-Code man unter `quadrant.c` ins Verzeichnis `serie02`.

Aufgabe 10*. Unter einer **rekursiven Funktion** versteht man eine Funktion, die sich selber aufruft. Das folgende Beispiel ist Ihnen allen bekannt: Für $n \in \mathbb{N}_0$ definieren wir $f(0) = 1$ und $f(n) := n \cdot f(n-1)$ für $n \geq 1$. Um welche Funktion handelt es sich? Man schreibe sich dazu beispielsweise alle Faktoren von $f(5)$ hin: $f(5) = 5 \cdot f(4) = 5 \cdot 4 \cdot f(3) \ldots$. Implementieren Sie diese rekursive Funktion und schreiben Sie ein aufrufendes Hauptprogramm, das $n \in \mathbb{N}_0$ einliest und $f(n)$ ausgibt. Wählen Sie einen geeigneten Namen für den Source-Code und speichern Sie ihn ins Verzeichnis `serie02`.

Aufgabe 11*. Man schreibe eine rekursive Funktion `binomial`, die den Binomialkoeffizienten $\binom{n}{k}$ berechnet. Dazu verwende man das Additionstheorem $\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$. Ferner schreibe man eine Funktion `binomial2`, die den Binomialkoeffizienten mittels $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ berechnet. Dazu ist eine rekursive Funktion `faktorielle` zu entwickeln, die zu gegebenem $n \in \mathbb{N}$ die Faktorielle $n!$ berechnet. Man schreibe ein aufrufendes Hauptprogramm, in dem $k, n \in \mathbb{N}_0$ mit $k \leq n$ eingelesen und $\binom{n}{k}$, berechnet auf beide Weisen, ausgeben werden. Den Source-Code speichere man unter `binomial.c` ins Verzeichnis `serie02`.

Aufgabe 12. Gegeben seien $x, y \in \mathbb{N}$. Man schreibe eine rekursive Funktion `test`, die herausfindet, ob $x = y^n$ für ein geeignetes $n \in \mathbb{N}$ gilt. In diesem Fall werde $n \in \mathbb{N}$ zurückgegeben, anderenfalls Null. Man schreibe ein aufrufendes Hauptprogramm, dass x und y einliest und das Ergebnis ausgibt.

Aufgabe 13. Die Fibonacci-Folge ist definiert durch $x_0 := 0$, $x_1 := 1$ und $x_{n+1} := x_n + x_{n-1}$. Man schreibe eine rekursive Funktion `fibonacci`, die zu gegebenem Index k das Folgenglied x_k zurückgibt. Ferner schreibe man ein aufrufendes Hauptprogramm, as den Index n einliest und x_n ausgibt.

Aufgabe 14. Gegeben Sei eine stetige Funktion $f : [a, b] \to \mathbb{R}$ auf einem Intervall $[a, b]$. Es gelte

$$f(a) \cdot f(b) \leq 0.$$

Dann hat f eine Nullstelle x_0, die im Folgenden mittels Bisektion approximiert werden soll: Der Bisektionsalgorithmus arbeitet wie folgt: Man definiert $c := (a + b)/2$ als Intervallmittelpunkt. Aufgrund der Voraussetzung gilt

$$f(a) \cdot f(c) \leq 0 \quad \text{oder} \quad f(c) \cdot f(b) \leq 0.$$

Im Fall $f(a) \cdot f(c) \leq 0$ ruft man den Bisektionsalgorithmus für das Intervall $[a, c]$ auf, anderenfalls für das Intervall $[c, b]$. Als Abbruchbedingung verwende man $|b - a| \leq \varepsilon$, d.h. eine Bedingung an die Intervallbreite. Wegen $x_0 \in [a, b]$ ist dann beispielsweise a eine Approximation der Nullstelle bis auf einen Fehler ε. Der Bisektionsfunktion `bisektion` werden also die Parameter $a, b \in \mathbb{R}$ und $\varepsilon > 0$ übergeben. Im Fall $f(a) \cdot f(b) > 0$ soll eine Fehlermeldung ausgegeben werden. Ansonsten sollen bei jedem Funktionsaufruf a, b, $|b - a|$ und $f(a)$ ausgegeben werden. Als Testfunktion verwende man $f(x) = x^2 + \exp(x) - 2$ auf $[0, \infty)$, die man als eigene C-Funktion realisiere. Man schreibe ferner ein Hauptprogramm, das $b, \varepsilon > 0$ eingeliest und die Approximation von x_0 ausgibt.

Aufgabe 15. Gegeben seien zwei Geraden $f(x) = ax + b$ und $g(x) = cx + d$. Man schreibe eine Prozedur `schnittpunkt`, die überprüft, ob sich die beiden Geraden f und g schneiden und ggf. den Schnittpunkt berechnet. Das Ergebnis soll geeignet ausgegeben werden. Ferner schreibe man ein Hauptprogramm, in dem die Parameter $a, b, c, d \in \mathbb{R}$ eingelesen werden.

Aufgabe 16. Gegeben seien drei Punkte (x, y), (u, v) und (a, b) in $\mathbb{R}^2$. Man schreibe eine Funktion `punkte`, die überprüft, ob die 3 Punkte auf einer Geraden liegen. Ferner schreibe man ein aufrufendes Hauptprogramm, in dem die 6 Parameter eingelesen werden und das Resultat ausgegeben wird.