

## Übungen zu Algorithmen

Wintersemester 2003/2004

### Blatt 3

#### Aufgabe 3.1 (30 Punkte)

Schreiben Sie ein Java-Programm, welches zwei Zahlen vom Typ `int` als untere und obere Grenzen einliest und alle Primzahlen, die zwischen diesen liegen, ausgibt. Erzwingen Sie für die Obergrenze die Eingabe einer Zahl, die größer ist als die Untergrenze.

#### Aufgabe 3.2 (10 Punkte)

Überführen Sie die folgenden Zahlen in die 4-Bit Zweier-Komplement-Darstellung. Führen Sie die arithmetische Operation im Zweier-Komplement durch, und wandeln Sie das Ergebnis in die dezimale Darstellung um. Überprüfen Sie das Ergebnis mit dem Verfahren der Verdoppelung des führenden Bits.

a)  $5 + 7$

b)  $3 - 7$

c)  $-1 - 4$

#### Aufgabe 3.3 (10 Punkte)

Betrachten Sie das Java-Programm `/home/ainf/Uebung/Blatt3/Aufg3.3.java`. Kompilieren Sie das Programm und führen Sie es aus. Erklären Sie ihrem Tutor **schriftlich** wie die Ergebnisse der Berechnungen zustande kommen.

#### Aufgabe 3.4 (15 Punkte)

Lösen Sie die folgenden Aufgaben **schriftlich**.

a) Geben Sie die 32-Bitcodierung nach dem IEEE-Standard 754-1985 für die drei folgenden Gleitkommazahlen vom Typ `float` an:

i) 68.375

ii) -0.2509765625

iii) 0.8

b) Welche Zahl wird codiert durch:

0 10000100 010100000000000000000000

s Exponent                      Mantisse

Achtung: Benutzen Sie einen "Taschenrechner" mit ausreichender Genauigkeit.

#### Aufgabe 3.5 (15 Punkte)

Lösen Sie **schriftlich** die folgenden Aufgaben.

a) Wie werden Gleitkommazahlen addiert und multipliziert?

b) Beantworten Sie die folgenden Fragen für die Darstellungen von `float` **und** `double` im IEEE-Standard 754-1985.

- i) Wie lautet Codierung und Wert der kleinsten positiven Zahl in normalisierter Darstellung?
- ii) Wie viele verschiedene numerische Werte können dargestellt werden?
- iii) Wandeln Sie die Zahl 1.000.000.000 jeweils in die 32 Bit Float und in die 64 Bit Double Darstellung um. Wie lauten jeweils Codierung und Wert der drei in Bit-Darstellung folgenden Werte?

**Aufgabe 3.6 (20 Punkte)**

Schreiben Sie ein Java-Programm, welches die Glieder ( $p_n$ ) einer Folge bestimmt, deren Grenzwert  $\sqrt{2}$  ist. Geben Sie ein Folgeglied aus, dessen Wert sich nur wenig von  $\sqrt{2}$  unterscheidet. Die Folge ist wie folgt definiert:

$$p_1 = 2, \quad p_{n+1} = \frac{1}{2} \left( p_n + \frac{2}{p_n} \right), \quad n \geq 1$$

Ihr Programm sollte terminieren, wenn  $|p_n^2 - 2| < \epsilon$  erreicht ist. Achtung: Wählen Sie  $\epsilon$  nicht zu klein. Was passiert bei zu kleinem  $\epsilon$  ?

**Aufgabe 3.7 (0 Punkte)**

Machen Sie sich mit den folgenden UNIX-Kommandos vertraut:

Kommando	Erklärung
<code>cd ../../ainf/Uebung/Blatt2</code>	Wechselt vom Heimatkatalog in den Musterlösungskatalog.
<code>cd /home/ainf/Uebung/Blatt2</code>	Wechselt in den Musterlösungskatalog.
<code>cd ~</code>	Wechselt in den Heimatkatalog.
<code>cp Aufg2_3.java ~</code>	Kopiert die Datei <code>Aufg2_3.java</code> in den Heimatkatalog.
<code>cp Aufg2_3.java ~/Blatt2</code>	Kopiert <code>Aufg2_3.java</code> in den eigenen Katalog <code>Blatt2</code> .

Ihr Tutor würde sich freuen, wenn Sie ihm folgende *vi*-Kommandos vorführen könnten:

Mode	Kommando	Erklärung
K	0	Setzt den Cursor an den Zeilenanfang.
K	\$	Setzt den Cursor an das Zeilenende.
K	w, W	Bewegt den Cursor nach rechts bis zum nächsten Wortanfang ( <i>word</i> ).
K	b, B	Bewegt den Cursor nach links bis zum nächsten Wortanfang ( <i>back</i> ).
K	e, E	Bewegt den Cursor nach rechts bis zum Wortende ( <i>end</i> ).
K	f <i>x</i>	Setzt den Cursor auf das nächste <i>x</i> rechts von der aktuellen Position ( <i>find</i> ).
K	F <i>x</i>	Setzt den Cursor auf das nächste <i>x</i> links von der aktuellen Position ( <i>find</i> ).
K	;	Wiederholt das letzte f- oder F-Kommando.
K	t <i>x</i>	Setzt den Cursor vor das nächste <i>x</i> rechts von der aktuellen Position ( <i>to</i> ).
K	T <i>x</i>	Setzt den Cursor vor das nächste <i>x</i> links von der aktuellen Position ( <i>to</i> ).
K	%	Setzt den Cursor auf die korrespondierende Klammer.
K	~	Der Buchstabe unter dem Cursor wird groß bzw. klein.
K	>>	Die Zeile um einen Tabulator nach rechts shiften.
K	<<	Die Zeile um einen Tabulator nach links shiften.
K	.	Das letzte Kommando wiederholen.
Z	r header	Liest die Datei <code>header</code> nach der aktuellen Zeile ein ( <i>read</i> ).

**PS.:** Die Musterlösungen der Übungsaufgaben finden Sie im Verzeichnis `/home/ainf/Uebung/Blattn`. (*n* ist die Nummer des abgeschlossenen Übungsblatts.)

Die in der Vorlesung besprochenen Java-Programme finden Sie im Verzeichnis `/home/ainf/Vorlesung`.