Prof. H.-Chr. Grunau
Dipl.-Math. L. Pulst

08.10.2012
Wintersemester 2012/13

ÜBUNGEN ZUR ALGORITHMISCHEN MATHEMATIK
BLATT 1

(1) Melden Sie sich am Computer an.

(2) Öffnen Sie den Editor *"KWrite"* und die Konsole.

(3) Geben Sie folgendes Programm in den Editor ein (Additionsmaschine, 1.1 in der Vorlesung) und kompilieren Sie das Programm.

```
#include <stdio.h>

int main(void)
{
 int zahl1,zahl2,ergebnis;

 printf("Dieses Programm kann zwei ganze Zahlen addieren.\n\
 Geben Sie bitte Ihren ersten Summanden ein: ");

 scanf("%i",&zahl1);

 printf("Geben Sie nun bitte Ihren zweiten Summanden ein: ");

 scanf("%i",&zahl2);

 ergebnis=zahl1+zahl2;

 printf("Die Summe von %i und %i lautet: %i.\n\n",zahl1,zahl2,ergebnis);

 return(0);
}
```

(4) Führen Sie das Programm aus und testen Sie sehr große ganze Zahlen (Betrag größer als 2^{31}). Ändern Sie das Programm so, dass es auch mit noch größeren ganzen Zahlen funktionieren kann.

(5) Informieren Sie sich über den Variablentyp `char` und schreiben Sie ein C-Programm, welches zu einem eingelesenen Zeichen den Wert in der ASCII-Tabelle ausgibt.

(6) Ist der folgende Quelltextteil syntaktisch korrekt?

```
 int diff=c-5;
```

(7) Schreiben Sie ein C-Programm, welches zehn nacheinander eingegebene Zeichen einliest, und diese danach wieder ausgibt.

Hinweis: Der Befehl `getchar`.

(8) Kann man (7) vereinfachen?

Hinweis: Felder von Variablen: `Arrays`.

Die Beispielprogramme von Prof. Grunau finden Sie unter
http://www-ian.math.uni-magdeburg.de/home/grunau/additional.html.

Die Aufgabenblätter finden Sie unter
http://www-ian.math.uni-magdeburg.de/∼pulst/Algo_Math.html.

Prof. H.-Chr. Grunau
Dipl.-Math. L. Pulst

15.10.2012
Wintersemester 2012/13

ÜBUNGEN ZUR ALGORITHMISCHEN MATHEMATIK
BLATT 2

(1) Schreiben Sie ein C-Programm, welches vier Zeichen einliest und danach prüft, ob ein fünftes eingegebenes Zeichen unter den ersten vier Zeichen war. Trifft dieses zu, soll dieses Zeichen getrennt von den anderen Zeichen ausgegeben werden.

Hinweis: Die Kontrollstruktur `if-else`.

(2) Dieses Programm soll nun schrittweise erweitert werden. Zunächst wollen wir Zeichenketten mit einer Länge von maximal 80 Zeichen einlesen. Das Zeichen für „Return“ ist als Eingabezeichen nicht zulässig, sondern soll als Zeichenkettenendzeichen verwendet werden. Sie sollten also mit `char-arrays` arbeiten und dabei den Laufindex mit Hilfe einer `for`-Schleife abarbeiten. Intern sollen Sie die Zahl 0 als Zeichenkettenendzeichen verwenden. Realisieren Sie mit Ihrem ersten Programm Ein- und Ausgabe solcher Zeichenketten. Sie sollten vorab das `char-array` zur Vermeidung von Abfragefehlern komplett mit Nullen initialisieren.

Hinweis: Sie müssen zunächst den ASCII-Code für „Return“ herausfinden.

(3) Für `char-arrays` führt man auch den Variablentyp `string` mit dem Formatzeichen `%s` ein. `printf` und `scanf` unterstützen diesen Variablentyp. Vereinfachen Sie damit das Programm aus (2). Welche Nachteile handeln Sie sich damit ein?

(4) ★ Erweitern Sie das in (2) erarbeitete Programm um die Funktionalität aus (1). Schreiben Sie anschließend ein zweites Programm mit der folgenden erweiterten Funktionalität: Der Nutzer soll ein Zeichenpaar eingeben, nach dem das Programm die ursprüngliche Zeichenkette durchsucht. Dieses Zeichenpaar, falls vorhanden, soll durch ein weiteres, vom Nutzer eingegebenes Zeichenpaar ersetzt werden. Nach solchen Ersetzungen soll mit den auf die ersetzten Paare folgenden Zeichen fortgefahren werden.

Hinweis: In der `if`-Anweisung kann man logische Verknüpfungen wie z.B. das „und“ `&&` und das „oder“ `||` einsetzen.

Hinweis zur Abgabe: Abweichend von der üblichen Regel reicht es beim zweiten Programm aus, wenn Sie es ausdruckt abgeben und per e-mail einsenden.

(5) ★ Schreiben Sie ein Programm, das die Dezimaldarstellung einer eingegebenen natürlichen Zahl in ihre Dualzifferndarstellung umwandelt. Die Ausgabe soll als Dualzahl erfolgen. Ihren Algorithmus müssen Sie ausführlich begründen.

Hinweis: Mit Hilfe des Operators `%` können Sie den Rest bei der Division zweier natürlicher Zahlen bestimmen.

Die mit ★ gekennzeichneten Aufgaben sind Hausaufgaben. Abgabe der Lösungen: 22.10.2012 vor der Vorlesung.

Bitte beachten Sie die Hinweise zur Abgabe von Programmieraufgaben: Grundsätzlich müssen Sie Programme handschriftlich und per e-mail abgeben. Ihre handschriftliche Lösung muss das Programm angemessen dokumentieren.

Die übrigen Aufgaben sind Präsenzaufgaben für die Übungen am 16./17.10.

Prof. H.-Chr. Grunau
Dipl.-Math. L. Pulst

22.10.2012
Wintersemester 2012/13

ÜBUNGEN ZUR ALGORITHMISCHEN MATHEMATIK
BLATT 3

(1) Ersetzen Sie die geschachtelten `if - else`-Anweisungen in dem Programm „Taschenrechner1.c“ aus der Vorlesung durch `switch - case`. Die Wahl der Rechenart soll mit `a-s-m-d` gesteuert werden.

(2) (a) ★Schreiben Sie ein Programm, welches die Fakultätsfunktion rekursiv berechnet, d.h. für die Berechnung von $(n-1)!$ in $n! = n(n-1)!$ soll die Funktion selbst wieder aufgerufen werden.

(b) ★Sei $n \in \mathbb{N}$ gerade. Beweisen Sie folgende Ungleichung

$$n! \geq \left(\frac{n}{2}\right)^{\frac{n}{2}}.$$

Hinweis: Ordnen Sie $n! = n(n-1)(n-2) \cdot \ldots \cdot 3 \cdot 2 \cdot 1$ geschickt um.

(3) Ziel dieser Aufgabe ist es, das *Sieb des Eratosthenes* zu programmieren. Das Sieb findet alle Primzahlen bis zu einer vorgegebenen Grenze $N \in \mathbb{N}$. Dafür stelle man sich vor, dass alle Zahlen bis N aufgelistet sind. Zuerst wird die Eins gestrichen, sie ist keine Primzahl. Danach werden alle Vielfachen der Zwei gestrichen, da diese keine Primzahlen sein können. Auch alle Vielfachen der Drei werden als zusammengesetzte Zahlen gestrichen. Diese Prozedur wird mit allen Vielfachen der nächsten ungestrichenen (Prim-)Zahl kleiner gleich N wiederholt. Übrig bleiben nur noch die Primzahlen.

(a) Beweisen Sie, dass es ausreicht die Vielfachen aller Primzahlen kleiner oder gleich $\sqrt{N}$ zu streichen, um alle Primzahlen bis $N \in \mathbb{N}$ auszusieben.

(b) Programmieren Sie das Sieb des Eratosthenes. Listen Sie dazu die natürlichen Zahlen in einem `array` der Maximallänge auf, die Ihr Computer Ihnen ermöglicht, d.h. Ihr Sieb soll auch nur alle Primzahlen bis zu dieser Grenze finden können.

(c) Erweitern Sie Ihr Programm aus (b) so, dass es auch alle Primzahlzwillinge, -drillinge und -vierlinge findet.

Hinweis: Primzahlzwillinge sind zwei aufeinanderfolgende Primzahlen p_n und p_{n+1} mit $p_{n+1} - p_n = 2$. Primzahldrillinge sind Primzahlen von der Form $p,\ p+2$ und $p+6$ sowie der spezielle Drilling $3,\ 5$ und 7. Entsprechend sind Primzahlvierlinge Primzahlen von der Form $p,\ p+2,\ p+6$ und $p+8$.

(4) ★ Schreiben Sie ein naives Programm, welches zu zwei natürlichen Zahlen den *ggT* (größten gemeinsamen Teiler) findet.

(5) Werten Sie mit Hilfe einer `for`-Schleife das Polynom

$$10^9 \cdot (x^2 - 100)^4 - 16322408010$$

in dieser Form und ausmultipliziert, d.h.

$$10^9 \cdot (x^8 - 400x^6 + 60000x^4 - 4000000x^2 + 100000000) - 16322408010,$$

in der Nähe von 10 aus.

Hinweis: `double`-Konstanten werden in der Form `400.0` geschrieben.

Achtung! Die Übungen vom 31.10.12 werden auf folgende Termine verschoben:

Prof. Grunau: Dienstag, 30.10.12, 19.00 - 20.30 Uhr, G02-112;
L. Pulst: Donnerstag, 01.11.12, 7.30 - 9.00 Uhr, G02-112.

Sollten Sie zum Zeitpunkt Ihrer Übung verhindert sein, können Sie in dieser Woche auch die Übung wechseln.

Die mit ★ gekennzeichneten Aufgaben sind Hausaufgaben. Abgabe der Lösungen: 29.10.2012 vor der Vorlesung.

Bitte beachten Sie die Hinweise zur Abgabe von Programmieraufgaben: Grundsätzlich müssen Sie Programme handschriftlich und per e-mail abgeben. Ihre handschriftliche Lösung muss das Programm angemessen dokumentieren.

Die übrigen Aufgaben sind Präsenzaufgaben für die Übungen am 23./24.10.

Prof. H.-Chr. Grunau
Dipl.-Math. L. Pulst

29.10.2012
Wintersemester 2012/13

ÜBUNGEN ZUR ALGORITHMISCHEN MATHEMATIK
BLATT 4

(1) Wir nehmen das Programm zum Einlesen von Zeichenketten aus Aufgabe (2) von Blatt 2 wieder auf.

(a) Zunächst soll die Eingabe dahingehend vereinfacht werden, dass Sie alle Zeichen insgesamt eintippen und danach die Eingabe mit return abschließen. Das Programm soll mit einer Fehlermeldung beendet werden, wenn Sie mehr als 80 Zeichen eingegeben haben. Falls Sie maximal 80 Zeichen eingegeben haben, sollen diese wieder ausgegeben werden.

(b) ★ Schreiben Sie ein Programm, das Sie auffordert, eine natürliche Zahl zwischen 0 und 4 294 967 295 einzugeben, Ihre Eingabe als Zeichenkette einliest und auf Zulässigkeit überprüft. Im Falle einer unzulässigen Eingabe soll das Programm mit einer Fehlermeldung beendet werden. Im Falle einer zulässigen Eingabe soll die Zeichenkette als Integervariable dargestellt und als solche ausgegeben werden.

Hinweis. Verwenden Sie die `isdigit`-Funktion aus der Headerdatei ctype.h.

(2) ★ Berechnen Sie die Dualzahldarstellung von $\frac{1}{10}$, d.h. die Koeffizienten a_i in

$$\frac{1}{10} = \sum_{i=1}^{\infty} a_i 2^{-i}, \qquad a_i \in \{0,1\}.$$

Berücksichtigen Sie jetzt nur die ersten 32 gültigen Dualziffern. D.h., sei i_0 der kleinste Index i mit $a_i = 1$; berechnen Sie

$$\sum_{i=i_0}^{i_0+31} a_i 2^{-i} \quad \text{bzw.} \quad \left(\sum_{i=i_0}^{i_0+31} a_i 2^{-i} \right) - \frac{1}{10}.$$

Das zeigt: Schon bei der Umwandlung von endlichen Dezimalbrüchen in approximierende endliche Dualbrüche treten Rundungsfehler auf!

Hinweis. Sie dürfen ohne Beweis verwenden, dass für $|q| < 1$ gilt: $\sum_{i=0}^{\infty} q^i = \frac{1}{1-q}$. Sollte Ihnen das Rechnen mit Reihen unheimlich sein, schreiben Sie 10 als Dualzahl `1010`, und dividieren Sie `1` im Dualsystem schriftlich durch diese Zahl solange, bis Sie 32 gültige Dualziffern haben. Berechnen Sie den Wert dieses Dualbruchs und den Unterschied zu $\frac{1}{10}$.

(3) Rationale Zahlen könnte man natürlich als Gleitkommazahlen implementieren, handelt sich aber damit die bereits ausfürlich thematisierte Rundungsfehlerproblematik ein. Wir beschränken uns auf positive rationale Zahlen $\frac{a}{b}$, $(a, b \in \mathbb{N})$: Solange wir nicht mit der Überlaufproblematik konfrontiert werden, können wir diese als Zahlenpaar (a, b) mit Integervariablen a und b exakt abspeichern. Sie sollen einige C-Programme zur Bruchrechnung entwickeln.

(a) *Kürzen:* Schreiben Sie ein Programm, das den Bruch $\frac{a}{b}$, $(a, b \in \mathbb{N})$, in gekürzter Darstellung ausgibt.

*Hinweis.*Sie können ein ggT-Programm von Blatt 3 verwenden.

(b) *Addieren:* Schreiben Sie ein Programm, das die Brüche $\frac{a}{b}$ und $\frac{c}{d}$, $(a, b, c, d \in \mathbb{N})$, addiert. Dabei soll immer erst gekürzt werden; die Addition soll mit Hilfe des Hauptnenners erfolgen, und auch das Ergebnis soll in gekürzter Darstellung ausgegeben werden. Das Programm soll alle Rechenschritte ausgeben.

Hinweis. Zur Hauptnennerbestimmung benötigen Sie das kleinste gemeinsame Vielfache $\mathrm{kgV}(a, b)$ zweier natürlicher Zahlen a und b. Sie dürfen ohne Beweis verwenden, dass $\mathrm{kgV}(a, b) = \frac{ab}{\mathrm{ggT}(a,b)}$.

(c) Modifizieren Sie das vorhergehende Programm dahingehend, dass nur diejenigen Rechenschritte ausgegeben werden, die sich vom jeweils vorhergehenden unterscheiden.

(4) ★ Die Fibonaccizahlen sind rekursiv definiert: $a_0 := 1$, $a_1 := 1$, für $k \geq 2$:

$$a_k := a_{k-1} + a_{k-2}.$$

(a) Schreiben Sie zwei Programme zur Berechnung der Fibonaccizahlen, eines durch rekursiven Aufruf einer entsprechend Funktion, eines unter Vermeidung desselben.

(b) Zeigen Sie, dass für alle $k \in \mathbb{N}_0$ gilt:

$$a_{2k} \geq 2^k, \qquad a_{2k+1} \geq 2^k.$$

Achtung! Wegen des Feiertags können die Übungen am Mittwoch, den 31.10.12 nicht stattfinden. Ersatztermine:

Prof. Grunau: Dienstag, 30.10.12, 19.00 - 20.30 Uhr, G02-112;
L. Pulst: Donnerstag, 01.11.12, 7.30 - 9.00 Uhr, G02-112.

Sollten Sie zum Zeitpunkt Ihrer Übung verhindert sein, können Sie in dieser Woche auch die Übung wechseln.

Die mit ★ gekennzeichneten Aufgaben sind Hausaufgaben. Abgabe der Lösungen jeweils vor der Vorlesung:

Aufgabe 4*: 05.11.2012,* **Aufgaben 1b) und 2***: 12.11.2012.*

Bitte beachten Sie die Hinweise zur Abgabe von Programmieraufgaben: Grundsätzlich müssen Sie Programme handschriftlich und per e-mail abgeben. Ihre handschriftliche Lösung muss das Programm angemessen dokumentieren.

Die übrigen Aufgaben sind Präsenzaufgaben für die Übungen am 30.10/01.11. und 6./7.11.

Der Ort für die Klausur am 28.01.13, 9.00 Uhr - 11.00 Uhr, steht fest: G05-H4.

Prof. H.-Chr. Grunau
Dipl.-Math. L. Pulst

12.11.2012
Wintersemester 2012/13

ÜBUNGEN ZUR ALGORITHMISCHEN MATHEMATIK
BLATT 5

(1) Eine Zahl $v \in \mathbb{Z}$ heißt *kleinstes gemeinsames Vielfaches* zweier Zahlen $a, b \in \mathbb{Z} \setminus \{0\}$, $v = \mathrm{kgV}(a, b)$, falls gilt:
  (i) $a \,|\, v$ und $b \,|\, v$;
  (ii) ist $c \in \mathbb{Z}$ eine weitere Zahl mit $a \,|\, c$ und $b \,|\, c$, so folgt $v \,|\, c$;
  (iii) $v \in \mathbb{N}$.
Es wird $\mathrm{kgV}(a, b) = 0$ gesetzt, falls $a = 0$ oder $b = 0$.

Zeigen Sie für alle $a, b \in \mathbb{Z} \setminus \{0\}$:

$$\mathrm{kgV}(a, b) \cdot \mathrm{ggT}(a, b) = |a \cdot b|.$$

(2) ★ Seien $a_0 := 1,\ a_1 := 1$ und für $k \geq 2$:

$$a_k := a_{k-1} + a_{k-2}$$

die Fibonaccizahlen aus Aufgabe 4, Blatt 4.

(a) Zeigen Sie, dass der euklidische Algorithmus bei Eingabe von $a = a_{k+2},\ b = a_{k+1},\ k \in \mathbb{N}_0$, genau $k + 1$ Divisionen ausführt.

(b) Zeigen Sie per vollständiger Induktion für $k \in \mathbb{N}_0$

$$a_k = \frac{1}{\sqrt{5}} \left(\left(\frac{1 + \sqrt{5}}{2} \right)^{k+1} - \left(\frac{1 - \sqrt{5}}{2} \right)^{k+1} \right).$$

Hinweis. Zeigen Sie, dass die Lösungen von $\phi^2 = \phi + 1$ die Gleichung $\phi^i = \phi^{i-1} + \phi^{i-2}$ für alle $i \geq 2$ erfüllen.

(3) (a) Schreiben Sie ein Programm, welches Ihnen die Anzahl der benötigten Divisionen sowie die Anzahl der Divisionen mit $q_i = 1$ bzw. $q_i = 2$ im Euklidischen Algorithmus zurürck gibt. q_i bezeichnet hier den Quotienten wie im Beweis zu Satz 2.5 aus der Vorlesung.

(b) Schreiben Sie nun ein Programm, welches den Euklidischen Algorithmus automatisch auf Zahlenpaare (a, b) mit $a, b \in \mathbb{N}$ und $a, b \leq M$ anwendet. Der Maximalwert M soll vom Benutzer eingegeben werden können. Das Programm soll auch den Prozentsatz der Divisionen mit $q_i = 1$ bzw. $q_i = 2$ berechnen und die maximale Anzahl an benötigten Divisionen ausgeben, die während der Berechnungen bei einem Durchgang des Euklidischen Algorithmus auftritt.

(c) ★ Schreiben Sie ein Programm, welches experimentell die relative Häufigkeit für die Ereignisse: „$\mathrm{ggT}(a, b) = 1$“ bzw. „$\mathrm{ggT}(a, b) \leq 10$“ feststellt. Wählen Sie dafür $a, b \in \mathbb{N}$ und $a, b \leq M$. Der Maximalwert M soll wieder vom Benutzer eingegeben werden können.

Die mit ★ gekennzeichneten Aufgaben sind Hausaufgaben. Abgabe der Lösungen: 19.11.2012 vor der Vorlesung.

Bitte beachten Sie die Hinweise zur Abgabe von Programmieraufgaben: Grundsätzlich müssen Sie Programme handschriftlich und per e-mail abgeben. Ihre handschriftliche Lösung muss das Programm angemessen dokumentieren.

Die übrigen Aufgaben sind Präsenzaufgaben für die Übungen am 13./14.11.

Prof. H.-Chr. Grunau
Dipl.-Math. L. Pulst

19.11.2012
Wintersemester 2012/13

ÜBUNGEN ZUR ALGORITHMISCHEN MATHEMATIK
BLATT 6

(1) Zeigen Sie, dass der euklidische Algorithmus bei der Eingabe $a > b > 0$ höchstens $c \log(b\sqrt{5})$ Divisionen benötigt, wobei $c = \left(\log \frac{1+\sqrt{5}}{2}\right)^{-1} \approx 2.08$. Dabei bezeichnen wir mit log den natürlichen Logarithmus.

Hinweis. Blatt 5, Aufgabe 2.

(2) Zu Zahlen $a_0, a_1, \ldots, a_n \in \mathbb{Z}$ mit $a_i \geq 1,\ i \geq 1$, ist der zugehörige *Kettenbruch* die rationale Zahl

$$[a_0; a_1, \ldots, a_n] = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ddots + \cfrac{1}{a_{n-1} + \cfrac{1}{a_n}}}}}.$$

(a) Sei $x \in \mathbb{Q}$. Zeigen Sie: Es existieren $a_0, a_1, \ldots, a_n \in \mathbb{Z}$ mit $a_i \geq 1,\ i \geq 1$, so dass $x = [a_0; a_1, \ldots, a_n]$.

Hinweis. Euklidischer Algorithmus mit den Eingabewerten a und b, wobei $x = \frac{a}{b}, a \in \mathbb{Z}, b \in \mathbb{N}$.

(b) Schreiben Sie ein Programm, welches zu einer gegebenen nichtnegativen rationalen Zahl die Kettenbruchdarstellung in der Form $[a_0; a_1, \ldots, a_n]$ berechnet. Die nichtnegative rationale Zahl $\frac{a}{b}$ soll als Zahlenpaar $(a, b),\ a \in \mathbb{N}_0, b \in \mathbb{N}$, eingegebenen werden.

(c) ★ Schreiben Sie ein Programm, welches zu einem Kettenbruch $[a_0; a_1, \ldots, a_n]$ mit $a_0 \geq 0$ die entsprechende rationale Zahl in gekürzter Darstellung ausgibt. Es dürfen nur `integer`-Variablen benutzt werden.

Hinweis. Auf Grund der **Überlaufproblematik** müssen sie damit rechnen ein falsches Ergebnis zu produzieren. Um Ihrem Ergebnis zu trauen, führen Sie per Hand mit Hilfe von (b) eine Probe durch. Diese soll in Teil (e) automatisiert werden.

Auf diese Aufgabe erhalten Sie bis zu 5 Punkte.

(d) ★ Erweitern Sie das Programm aus (b) mit folgender modifizierter Funktion aus (c): Nach der Ausgabe des Kettenbruchs $[a_0; a_1, \ldots, a_n]$ von $\frac{a}{b}$ soll die zu dem Kettenbruch $[a_0; a_1, \ldots, a_k],\ k \leq n$, gehörige rationale Zahl berechnet werden.

Hinweis. Bei der Abgabe dieser Aufgabe reicht es, wenn Sie den Code per e-mail und allein die Erläuterungen zu dem Programm handschriftlich abgeben. Auf diese Aufgabe erhalten Sie bis zu 5 Punkte.

(e) **(Zusatzaufgabe)** Fangen Sie bei den Berechnungen im Programm aus Teil c) einen möglichen Überlauf des Zahlbereichs wie folgt ab: Bevor Sie sich die berechnete rationale Zahl ausgeben lassen, soll das Programm wie in Teil b) den zugehörigen Kettenbruch berechnen. Die rationale Zahl ist korrekt, wenn der eingegebene und berechnete Kettenbruch in folgendem Sinne gleich sind: Ein Kettenbruch ist eindeutig bis auf die folgenden zwei Darstellungen:

$$[a_0; a_1, \ldots, a_{n-1}, a_n + 1] = [a_0; a_1, \ldots, a_{n-1}, a_n, 1].$$

Hinweis. Bei der Abgabe dieser Aufgabe reicht es, wenn Sie den Code per e-mail und allein die Erläuterungen zu dem Programm handschriftlich abgeben. Die Punkte auf diese Aufgabe werden Ihnen zusätzlich gutgeschrieben.

Bitte beachten Sie die zusätzlichen Hinweise zur Abgabe von Programmieraufgaben:

Ein-, Ausgabe- und Initialisierungsroutinen brauchen in der handschriftlichen Abgabe nur summarisch beschrieben zu werden, analog zur Darstellung in der Vorlesung.

Die mit ★ gekennzeichneten Aufgaben sind Hausaufgaben. Abgabe der Lösungen: 26.11.2012 vor der Vorlesung.

Bitte beachten Sie die Hinweise zur Abgabe von Programmieraufgaben: Grundsätzlich müssen Sie Programme handschriftlich und per e-mail abgeben. Ihre handschriftliche Lösung muss das Programm angemessen dokumentieren.

Die übrigen Aufgaben sind Präsenzaufgaben für die Übungen am 20./21.11.

Prof. H.-Chr. Grunau
Dipl.-Math. L. Pulst

26.11.2012
Wintersemester 2012/13

ÜBUNGEN ZUR ALGORITHMISCHEN MATHEMATIK
BLATT 7

(1) Seien $a_0 := 1$, $a_1 := 1$ und $a_k := a_{k-1} + a_{k-2}$ für $k \geq 2$ wieder die Fibonaccizahlen. Zeigen Sie, dass für alle $k \geq 1$ gilt:

(a) $a_{k+1}a_{k-1} - a_k^2 = (-1)^{k+1}$.

Hinweis. Blatt 5, Aufgabe 2.

(b) $a_k \cdot a_{k-1} \equiv (-1)^{k+1} \mod a_{k+1}$.

Insbesondere sind also a_k und a_{k-1} Einheiten in $\mathbb{Z}/a_{k+1}\mathbb{Z}$.

(2) ★ Berechnen Sie die letzten drei Dezimalstellen von 2^{1000}.

(3) Sei $p \in \mathbb{N}$ eine Primzahl. Ein Element $\overline{a} \in \mathbb{Z}/p\mathbb{Z}$ heißt *Primitivwurzel*, falls gilt:

$$\{\overline{a}^k : k = 0, 1, \ldots, (p-2)\} = \mathbb{Z}/p\mathbb{Z} \setminus \{\overline{0}\}.$$

(a) Schreiben Sie ein (naives) Programm, dass alle Primitivwurzeln in $\mathbb{Z}/p\mathbb{Z}$ berechnet.

(b) ★ In einem zweiten Schritt soll nun der Nutzer die Möglichkeit bekommen, eine dieser Primitivwurzeln $\overline{a} \in \mathbb{Z}/p\mathbb{Z}$ auszuwählen und ein Element $\overline{b} \in \mathbb{Z}/p\mathbb{Z}$ anzugeben. Ihr Programm soll eine Zahl $k \in \{0, 1, \ldots, (p-2)\}$ ermitteln, so dass gilt

$$\overline{a}^k = \overline{b} \quad \text{in} \quad \overline{b} \in \mathbb{Z}/p\mathbb{Z}.$$

Die mit ★ gekennzeichneten Aufgaben sind Hausaufgaben. Abgabe der Lösungen: 03.12.2012 vor der Vorlesung.

Bitte beachten Sie die Hinweise zur Abgabe von Programmieraufgaben. Bei Ihrer handschriftlichen Abgabe können Sie – analog zur Vorlesung – Standardinitialisierungs-, Ein- und Ausgabefunktionen summarisch darstellen.

Prof. H.-Chr. Grunau
Dipl.-Math. L. Pulst

03.12.2012
Wintersemester 2012/13

ÜBUNGEN ZUR ALGORITHMISCHEN MATHEMATIK
BLATT 8

(1) Sei $M := 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 = 9\,699\,690$. Zeigen Sie, dass für jedes $k \in \mathbb{Z}$ die Zahl m derjenigen Zahlen in $\{k+1, k+2, \dots, k+M\}$, die durch keine der Primzahlen $2, 3, 5, 7, 11, 13, 17, 19$ teilbar ist, gleich ist. Berechnen Sie m. Was kann man über das Verhältnis $\frac{m}{M}$ sagen, wenn man M durch weitere folgende Primzahlfaktoren vergrößert? Erweitern Sie das Programm aus Aufgabe 3, Blatt 3, dahingehend, dass Sie $\frac{m}{M}$ für beliebige Produkte aufeinanderfolgender Primzahlen unterhalb einer einzugebenden Grenze bestimmen können.

(2) ★ Betrachten Sie die Eulersche φ-Funktion, die für $m \in \mathbb{N}$, $m \geq 2$, die Anzahl der Elemente der Einheitengruppe von $\mathbb{Z}/m\mathbb{Z}$ angibt:

$$\varphi(m) := \#\{\overline{x} \in \mathbb{Z}/m\mathbb{Z} : \overline{x} \text{ ist Einheit in } \mathbb{Z}/m\mathbb{Z}\}.$$

(a) Sei $p \in \mathbb{N}$ eine Primzahl, $\ell \in \mathbb{N}$. Zeigen Sie:

$$\varphi(p^\ell) = p^\ell - p^{\ell-1}.$$

(b) Sei nun $m = p_1^{\ell_1} \cdot p_2^{\ell_2} \cdot \ldots \cdot p_k^{\ell_k}$ mit paarweise verschiedenen Primzahlen $p_1, p_2, \dots, p_k$. Zeigen Sie mit Hilfe des chinesischen Restsatzes:

$$\varphi(m) = \prod_{j=1}^{k} \left(p_j^{\ell_j} - p_j^{\ell_j - 1} \right) = m \prod_{j=1}^{k} \left(1 - \frac{1}{p_j} \right).$$

(3) ★ Betrachten Sie wie in der vorhergehenden Aufgabe die Eulersche φ-Funktion und schreiben Sie zwei Programme, die für zwei einzugebende Schranken $M_1, M_2 \in \mathbb{N}$, $2 \leq M_1 \leq M_2 \leq 2^{32}$, alle Werte der Eulerschen φ-Funktion

$$\varphi(m), \qquad m = M_1, \dots, M_2$$

berechnet.

(a) Bei dem ersten Programm greifen Sie direkt auf die Definition der φ-Funktion und das Einheitenkriterium Folgerung 3.10 der Vorlesung zurück.

(b) Bei Ihrem zweiten Programm implementieren Sie die Formel aus Aufgabe 2 (b).

Führen Sie einen experimentellen Laufzeitvergleich durch.

Die mit ★ gekennzeichneten Aufgaben sind Hausaufgaben. Abgabe der Lösungen: 10.12.2012 vor der Vorlesung.

Bitte beachten Sie die Hinweise zur Abgabe von Programmieraufgaben. Bei Ihrer handschriftlichen Abgabe können Sie – analog zur Vorlesung – Standardinitialisierungs-, Ein- und Ausgabefunktionen summarisch darstellen.

Prof. H.-Chr. Grunau
Dipl.-Math. L. Pulst

10.12.2012
Wintersemester 2012/13

ÜBUNGEN ZUR ALGORITHMISCHEN MATHEMATIK
BLATT 9

(1) (a) Schreiben Sie ein Programm welches prüft, ob die folgende Kongruenz

$$a \cdot x \equiv b \bmod m$$

in $\mathbb{Z}/m\mathbb{Z}$ lösbar ist. Die ganzen Zahlen $m > 1$, $m > a \geq 1$ und $m > b \geq 0$ sollen vom Nutzer eingegben werden können.

(b) Schreiben Sie das Programm `Chin_Restsatz.c` aus der Vorlesung so um, dass der Benutzer während des Ausführens festlegen kann, wieviele Kongruenzen er bearbeiten möchte.

Hinweis. Benutzen Sie dazu die Möglichkeit dynamische Arrays anzulegen, vgl. Sie dazu Paragraph 4.1 und das Programm `Einlesen_strings.c` aus der Vorlesung.

(c) ★ Schreiben Sie ein Programm welches die simultane Kongruenz

$$\begin{aligned} a_1 \cdot x &\equiv b_1 \bmod m_1 \\ \vdots \qquad & \qquad \vdots \\ a_k \cdot x &\equiv b_k \bmod m_k \end{aligned}$$

für $m = m_1 \cdot \ldots \cdot m_k$ in $\mathbb{Z}/m\mathbb{Z}$ löst. Die ganzen Zahlen $m_i > 1$, $m_i > a_i \geq 1$ und $m_i > b_i \geq 0$ für alle k sollen vom Benutzer eingegben werden können. Behandeln Sie das $k \in \mathbb{N}$ genauso dynamisch wie in Aufgabenteil (a), d.h. der Benutzer darf am Anfang des Programms das $k \in \mathbb{N}$ festlegen.

Hinweis. Kombinieren Sie die Aufgabenteile (a) und (b).

Bitte beachten Sie die folgenden Informationen zur Klausur:

- *Die Klausur findet am 04.02.2013 in G05-H4 von 9.15 Uhr - 10.45 Uhr (d.h. Dauer 90 Min.) statt. Bitte seien Sie pünktlich da.*
- *Bringen Sie bitte ausreichend Papier, Schreibzeug (Kugelschreiber oder Füller) und einen Lichtbildausweis mit.*
- *Während der Klausur dürfen Sie das* **gedruckte** *Vorlesungsskript von Prof. Grunau und ein festgebundenes Programmierhandbuch ihrer Wahl benutzen, allerdings keine losen Blätter-Sammlungen.*
- *Des weiteren sind keinerlei elektronische Hilfsmittel (Smartphones, Laptops, Netbooks, etc.) erlaubt.*

Die mit ★ gekennzeichneten Aufgaben sind Hausaufgaben. Abgabe der Lösungen: 17.12.2012 vor der Vorlesung.

Bitte beachten Sie die Hinweise zur Abgabe von Programmieraufgaben. Bei Ihrer handschriftlichen Abgabe können Sie – analog zur Vorlesung – Standardinitialisierungs-, Ein- und Ausgabefunktionen summarisch darstellen.

Prof. H.-Chr. Grunau
Dipl.-Math. L. Pulst

17.12.2012
Wintersemester 2012/13

ÜBUNGEN ZUR ALGORITHMISCHEN MATHEMATIK
BLATT 10

(1) Schreiben Sie das Programm `Einlesen_strings.c` aus Paragraph 4.1 der Vorlesung so um, dass alle Aktionen in klar definierte Funktionen ausgegliedert werden und das Hauptprogramm diese nur der Reihe nach aufruft.

(2) (a) Schreiben Sie ein Programm, welches eine „beliebig lange“ natürliche Zahl mit einer einstelligen natürlichen Zahl multipliziert. Speichern Sie dazu die „beliebig lange“ natürliche Zahl in ein Character-Feld, dessen Länge vorher vom Benutzer festgelegt werden kann.

(b) ★ Schreiben Sie ein Programm, welches zwei „beliebig lange“ natürliche Zahlen miteinander multipliziert. Die Länge der Zahlen soll vom Benutzer vorher bestimmt werden.

Hinweis. Benutzen Sie Teil (a) und das Verfahren der schriftlichen Multiplikation wie Sie es aus der Schule kennen.

Die Punkte auf diese Aufgabe werden Ihnen als **Zusatzpunkte** gutgeschrieben.

Die mit ★ gekennzeichneten Aufgaben sind Hausaufgaben. Abgabe der Lösungen: 07.01.2012 vor der Vorlesung.

Bitte beachten Sie die Hinweise zur Abgabe von Programmieraufgaben. Bei Ihrer handschriftlichen Abgabe können Sie – analog zur Vorlesung – Standardinitialisierungs-, Ein- und Ausgabefunktionen summarisch darstellen.

Fröhliche Weihnachten und einen guten Rutsch ins neue Jahr!

Prof. H.-Chr. Grunau
Dipl.-Math. L. Pulst

07.01.2013
Wintersemester 2012/13

ÜBUNGEN ZUR ALGORITHMISCHEN MATHEMATIK
BLATT 11

(1) Schreiben Sie ein Programm, das zwei „beliebig lange“ natürliche Zahlen als Listen speichert und Ihnen ausgibt, ob diese beiden Zahlen gleich sind, oder welche der beiden Zahlen die größere ist. Sie sollten dabei bei der Eingabe so vorgehen, dass führende Nullen vom Programm automatisch unterdrückt werden.

(2) ★ Sei $m \in \mathbb{N}$, $m \geq 2$. Ein Element $\overline{x}$ in $\mathbb{Z}/m\mathbb{Z}$ heißt *idempotent*, falls $x^2 \equiv x \mod m$.

(a) Sei zunächst $m = p^\ell$, $p \in \mathbb{N}$ eine Primzahl, $\ell \geq 1$. Man zeige, dass dann $\overline{0}$ und $\overline{1}$ genau die idempotenten Elemente in $\mathbb{Z}/m\mathbb{Z}$ sind.

Hinweis. Zeigen Sie zunächst: Gilt $x^2 \equiv x \mod p^\ell$ dann folgt: $p \mid x$ oder $p \mid (x-1)$.

(b) Sei nun $m = p_1^{\ell_1} \cdot p_2^{\ell_2} \cdot \ldots p_k^{\ell_k}$ mit paarweise verschiedenen Primzahlen $p_j \in \mathbb{N}$ und $\ell_j \geq 1$. Man zeige mit Hilfe des chinesischen Restsatzes: Es gibt in $\mathbb{Z}/m\mathbb{Z}$ genau 2^k idempotente Elemente. Benutzen Sie die Programme aus Übung und Vorlesung, um die idempotenten Elemente in $\mathbb{Z}/105\mathbb{Z}$ zu bestimmen.

(3) ★ Schreiben Sie das Programm `Primfaktorzerlegung.c` so um, dass für eine Zahl $m \geq 2$ mit der Primfaktorzerlegung $m = \prod_{j=1}^{k} p_j^{\ell_j}$ die gefundenen Primfaktoren p_j mitsamt ihren Vielfachheiten ℓ_j in einer Liste abgespeichert und entsprechend ausgegeben werden. Der j-te Knoten soll (p_j, ℓ_j) als Nutzdaten enthalten.

Hinweis zur handschriftlichen Abgabe. Programmteile, die Sie aus Beispielprogrammen aus der Übung oder der Vorlesung entnehmen, brauchen Sie nur summarisch darzustellen.

Die mit ★ gekennzeichneten Aufgaben sind Hausaufgaben. Abgabe der Lösungen: 14.01.2013 vor der Vorlesung.

Bitte beachten Sie die Hinweise zur Abgabe von Programmieraufgaben. Bei Ihrer handschriftlichen Abgabe können Sie – analog zur Vorlesung – Standardinitialisierungs-, Ein- und Ausgabefunktionen summarisch darstellen.

Bitte beachten Sie nochmals die folgenden Informationen zur Klausur:

- *Die Klausur findet am 04.02.2013 in G05-H4 von 9.15 Uhr - 10.45 Uhr (d.h. Dauer 90 Min.) statt. Bitte seien Sie pünktlich da.*
- *Bringen Sie bitte ausreichend Papier, Schreibzeug (Kugelschreiber oder Füller) und einen Lichtbildausweis mit.*
- *Während der Klausur dürfen Sie das* **gedruckte** *Vorlesungsskript von Prof. Grunau und ein festgebundenes Programmierhandbuch ihrer Wahl benutzen, allerdings keine losen Blätter-Sammlungen.*
- *Des weiteren sind keinerlei elektronische Hilfsmittel (Smartphones, Laptops, Netbooks, etc.) erlaubt.*

Prof. H.-Chr. Grunau
Dipl.-Math. L. Pulst

14.01.2013
Wintersemester 2012/13

ÜBUNGEN ZUR ALGORITHMISCHEN MATHEMATIK
BLATT 12

(1) Das Programm `Liste_doppelt_verkettet.c` aus der Vorlesung zum Einlesen „beliebig langer“ natürlicher Zahlen soll um Korrekturfunktionen ergänzt werden. Nach Eingabe der Ziffernkette soll diese angezeigt werden und der Nutzer Gelegenheit erhalten, diese nach einem vorgegebenen Zeichen zu durchsuchen.

   (a) In einem ersten Schritt soll der Nutzer Gelegenheit erhalten, dieses Zeichen an allen auftretenden Stellen durch jeweils ein anderes zu ersetzen bzw. beizubehalten. Nach jeder Ersetzung soll die Gelegenheit bestehen, den Vorgang zu beenden.

   (b) ★ Das in (a) erstellte Programm soll um die Möglichkeit erweitert werden, dass das jeweils gefundene Zeichen auch gelöscht werden kann.

   Hinweis. Führen Sie einen weiteren Hilfszeiger auf den jeweils aktuellen Knoten ein. Bedenken Sie, dass Sie beim Löschen des ersten bzw. letzten Zeichens auch den Listenanfangs- bzw. den Listenendknoten neu zuweisen müssen. Vergessen Sie nicht, den Speicher für den gelöschten Knoten freizugeben.

   Hinweis zur handschriftlichen Abgabe. Hier müssen Sie nur die Korrekturfunktion ausführlich darstellen.

(2) ★ Schreiben Sie analog zu Aufgabe 10.2 ein Programm, das zwei „beliebig lange“ natürliche Zahlen miteinander multiplizieren kann. Arbeiten Sie aber nun mit Listen.

Hinweis zur handschriftlichen Abgabe. Hier müssen Sie nur die Multiplikationsfunktion ausführlich darstellen; für die anderen Programmteile können Sie sich auf Beispielprogramme aus der Übung oder der Vorlesung beziehen.

Die mit ★ gekennzeichneten Aufgaben sind Hausaufgaben. Abgabe der Lösungen: 21.01.2013 vor der Vorlesung.

Bitte beachten Sie die Hinweise zur Abgabe von Programmieraufgaben.

Bitte beachten Sie nochmals die folgenden Informationen zur Klausur:

- *Die Klausur findet am 04.02.2013 in G05-H4 von 9.15 Uhr - 10.45 Uhr (d.h. Dauer 90 Min.) statt. Bitte seien Sie pünktlich da.*
- *Bringen Sie bitte ausreichend Papier, Schreibzeug (Kugelschreiber oder Füller) und einen Lichtbildausweis mit.*
- *Während der Klausur dürfen Sie das* **gedruckte** *Vorlesungsskript von Prof. Grunau und ein festgebundenes Programmierhandbuch ihrer Wahl benutzen, allerdings keine losen Blätter-Sammlungen.*
- *Des weiteren sind keinerlei elektronische Hilfsmittel (Smartphones, Laptops, Netbooks, etc.) erlaubt.*

Prof. H.-Chr. Grunau
Dipl.-Math. L. Pulst

21.01.2013
Wintersemester 2012/13

ÜBUNGEN ZUR ALGORITHMISCHEN MATHEMATIK
BLATT 13

(1) Erweitern Sie das Programm aus Aufgabe 1, Blatt 12, um eine Einfüge-Funktion, d.h. der Nutzer soll nach Eingabe der Ziffernkette zusätzlich zu den Funktionen „Ersetzen", „Löschen", „Weitersuchen" bzw. „Beenden" die Möglichkeit bekommen, eine beliebige Ziffer vor oder nach der gefundenen Ziffer einzufügen.

Bitte beachten Sie nochmals die folgenden Informationen zur Klausur:

- *Die Klausur findet am 04.02.2013 in G05-H4 von 9.15 Uhr - 10.45 Uhr (d.h. Dauer 90 Min.) statt. Bitte seien Sie pünktlich da.*
- *Bringen Sie bitte ausreichend Papier, Schreibzeug (Kugelschreiber oder Füller) und einen Lichtbildausweis mit.*
- *Während der Klausur dürfen Sie das* **gedruckte** *Vorlesungsskript von Prof. Grunau und ein festgebundenes Programmierhandbuch ihrer Wahl benutzen, allerdings keine losen Blätter-Sammlungen.*
- *Des weiteren sind keinerlei elektronische Hilfsmittel (Smartphones, Laptops, Netbooks, etc.) erlaubt.*