

## Übungsblatt 03

### E-Learning

Absolvieren Sie die Tests bis Di., 13.11., 23:55 Uhr.

Die Tests sind in der Stud.IP-Veranstaltung *Informatik I* unter *Lernmodule* hinterlegt.

Sie können einen Test **nur einmal durchlaufen**. Sobald Sie einen Test starten steht Ihnen nur eine **begrenzte Zeit** zu Verfügung, um den Test zu bearbeiten.

Alle Punkte, die Sie beim Test erreichen, werden ihnen angerechnet.

### ILIAS – 16 Punkte

#### Kodierung/Dekodierung

Nicht alle Anwendungen können mit den in der Vorlesung besprochenen Erweiterungen des ASCII-Zeichensatz arbeiten, z.B. der Kern des *Simple Mail Transfer Protocol* (SMTP). Damit diese Anwendungen trotzdem anderen Zeichensätzen, z.B. Texten mit Unicode-Zeichen, oder binäre Daten, z.B. Bilddateien, verarbeiten können werden diese Daten mit Hilfe (einer Auswahl) von ASCII-Zeichen codiert.

Informieren Sie sich **bevor Sie den Test starten** detailliert über die Codierungen *Quoted-Printable* und *Base64*.

Absolvieren Sie den Test *Informatik I - ILIAS 03 Teil 1*.

(16 Punkte)

#### Hinweis

Wenn Sie den Test einmal vollständig durchlaufen haben können Sie sich eine **Musterlösung** anzeigen lassen. Siehe dazu die Beschreibung unter ILIAS auf Übungsblatt 02.

### ILIAS 4-Minuten-Aufgaben – 12 Punkte

Absolvieren sie die Tests *Informatik I - ILIAS 03 Teil 2*, *Teil 3* und *Teil 4*.

(12 Punkte)

### LON-CAPA – 16 Punkte

#### Ablaufdiagramm und Java, Programmablauf verstehen

Absolvieren Sie im Test *Informatik I - LON-CAPA* die *Übung 03*.

(16 Punkte)

# Übung

**Abgabe bis Di., 13.11., 18 Uhr.**

Werfen Sie Ihre Lösung in den Zettelkästen Ihrer Gruppenübung. Für die Übungen im Nordbereich stehen die Zettelkästen im Sockelgeschoß (Ebene -1) **oder** auf dem Flur vor dem Seminarraum auf Ebene 0 des Instituts für Informatik.

Wenn Ihre Übung im Südbereich stattfindet, klären Sie mit Ihrem Tutor wo die Lösungen abzugeben sind.

Achten Sie darauf, dass Ihr **Name**, Ihre **Gruppe** und Ihre **Matrikelnummer** auf **jedem** Blatt stehen!

Falls Ihre Lösung mehrere Blätter umfasst, heften Sie diese bitte zusammen.

## Aufgabe 1 – 21 Punkte

### Basiskomplement

Übertragen Sie die Idee zur Darstellung negativer Zahlen mit dem Zweierkomplement in das Dezimalsystem.

1. Das Zweikomplement wird auch als Basiskomplement zur Basis 2 bezeichnet. Definieren Sie einen Datentyp **Gdec** (= „Göttinger decimal“) mit 8 Dezimalstellen, der zur Darstellung negativer Zahlen das Basiskomplement zur Basis 10 verwendet.

Der Datentyp soll folgende Eigenschaften haben.

- Die Anzahl negativer und nicht-negativer Zahlen sollte ungefähr gleich sein.
- Die Addition soll wie gewohnt (ziffernweise Addition mit Übertrag) durchführbar sein.
- Die Subtraktion kann auf Vorzeichenwechsel und Addition zurückgeführt werden.

Geben Sie genau an wie man für eine beliebige darstellbare ganze Zahl die 8 Dezimalstellen des Datentyps **Gdec** berechnet.

(13 Punkte)

#### Hinweise

- Überlegen Sie, was bei Dezimalstellen das Äquivalent zum *Kippen* eines Bits ist.
  - Die Summe einer positiven Zahl  $z$  in der üblichen Dezimaldarstellung und  $-z$  in Basiskomplementdarstellung zur Basis 10 ergibt 0.
2. Berechnen Sie folgende Terme mit dem Datentyp **Gdec**. Führen Sie die Subtraktion auf Addition und Vorzeichenwechsel zurück.
    - a)  $1789 - 1492$   
(4 Punkte)
    - b)  $1492 - 1789$   
(4 Punkte)

# Praktische Übung

Abgabe der Prüfsumme bis Di., 13.11., 23:55 Uhr.

Testat von Mi., 21.11., 8-10 Uhr bis Mo., 26.11., 18-20 Uhr.

Hilfe zum Bearbeiten der praktischen Übungen können Sie grundsätzlich jeden Tag in den Rechnerübungen bekommen, insbesondere in den Rechnerübungen **Di., 20.11., 8-10 Uhr und 18-20 Uhr**, in denen **keine Testate** stattfinden.

Lesen Sie die **Hinweise zu den praktischen Übungen**, die in der Stud.IP-Veranstaltung *Informatik I* unter *Dateien* → *Übungsblätter* hinterlegt sind.

- Erstellen Sie ein Archiv, dass **alle Dateien** enthält, die Sie beim Testat vorstellen möchten.
- Beim Testat werden nur Dateien aus einem Archiv testiert, dessen Prüfsumme **exakt** der von Ihnen übermittelten Prüfsumme entspricht.
- Berechnen Sie die Prüfsumme des Archivs mit dem **sha1sum** Befehl.
- Um die praktische Übung testieren zu lassen **müssen** Sie einen Termin in der zugehörigen Rechnerübung reservieren. Ein Testat ohne Termin ist nicht möglich. Testate zu einer anderen praktischen Übung können nur in Ausnahmefälle nach Rücksprache mit Herrn Brosenne durchgeführt werden.
- Übermitteln Sie die Prüfsumme durch Absolvieren des in der Stud.IP-Veranstaltung *Informatik I* unter *Lernmodule* hinterlegten Tests *Informatik I - ILIAS 03 Testat*.
- Öffnen Sie die Terminvergabe für diese praktische Übung und lassen Sie den von Ihnen reservierten Termin anzeigen.
- Entpacken Sie das Archiv erst nachdem der Tutor die Prüfsumme kontrolliert hat.

## Aufgabe 1 – 35 Punkte

### Middle Square

Ein Folge mit einfachen rekursiven Bildungsgesetz ist eine Auflistung von (unendlich vielen) fortlaufend nummerierten Folgengliedern (Zahlen), wobei ein Folgenglied (nur) aus dem unmittelbar vorhergehenden Folgenglied berechnet wird. Das erste Folgenglied (Startwert) wird explizit angegeben, somit können mit demselben Bildungsgesetz verschiedene Folgen aus unterschiedlichen Startwerten erzeugt werden.

#### Beispiel

Die Folge  $f_0, f_1, f_2, \dots$  mit Bildungsgesetz  $f_i = f_{i-1} + 2$  (für  $i > 0$ ) erzeugt mit Startwert  $f_0 = 2$  die Folge  $2, 4, 6, \dots$  der geraden Zahlen und mit Startwert  $f_0 = 1$  die Folge  $1, 3, 5, \dots$  der ungeraden Zahlen.

Eine Folge kann einen Zyklus, d.h. eine sich wiederholende minimale Teilfolge, enthalten. Eine Folge  $f_0, f_1, f_2, \dots$  enthält einen Zyklus  $g_1, \dots, g_n$  der Länge  $n$ , wenn es einen Index  $i \geq 0$  gibt, sodass Folgendes gilt.

$$f_0, f_1, f_2, \dots = f_0, \dots, f_{i-1}, \underbrace{g_1, \dots, g_n}_{\text{Zyklus}}, g_0, \dots, g_n, g_0, \dots, g_n, \dots$$

wobei  $f_i = g_1$  und  $g_1 \neq g_k$  für alle  $k = 2, \dots, n$

### Beispiel

Das Bildungsgesetz  $f_i = f_{i-1} \bmod 3 + 2$  (für  $i > 0$ ) erzeugt mit Startwert  $f_0 = 2$  die Folge  $2, 4, 3, 2, 4, 3, 2, 4, 3, 2, \dots$ . Diese Folge beginnt mit dem Zyklus  $2, 4, 3$  der Länge 3. Weiterhin sind die Zyklen  $4, 3, 2$  und  $3, 2, 4$  der Länge 3 in dieser Folge enthalten.

Die *Mittelquadratmethode* (*middle square method*) beschreibt ein einfaches rekursives Bildungsgesetz für Folgen. Informieren Sie sich über die *Mittelquadratmethode* unter <https://de.wikipedia.org>.

1. Programmieren Sie eine Java-Applikation, die einen *Middle Square Generator* für zweistellige Zahlen realisiert, wie folgt.

Dabei sollen, ausgehend von einem Startwert (dem ersten Folgenglied), die ersten 101 Folgenglieder der nach der *Mittelquadratmethode* gebildeten Folge für zweistellige Zahlen ausgegeben werden

- Initialisiere Sie eine `int`-Variable `a` durch interaktives Einlesen einer ganzen Zahl, negative Zahlen oder Zahlen mit mehr als zwei Stellen werden zurückgewiesen.

Benutzen Sie zum Einlesen die Klasse `StdIn` von *Sedgewick/Wayne*, die auch im Skript verwendet wird.

### Hinweis

Damit Sie Ihre Applikation compilieren und starten können, müssen Sie `StdIn.java` und `StdOut.java` von der Webseite <https://introcs.cs.princeton.edu/java/stdlib/> in das Verzeichnis mit dem Quellcode Ihrer Applikation kopieren.

- Iterieren Sie folgende Schritte 101-mal (durch Mitzaehlen, z.B. mit einer `for`-Schleife).
  - `a` ausgeben
  - `a` quadrieren
  - `a` aktualisieren, der neue Wert von `a` ist die zweistellige Zahl, die aus der Hunderter- und Zehnerstelle von `a` besteht
- Kennzeichnen Sie die Schritte *ausgeben*, *quadrieren*, *aktualisieren* durch Kommentare.

(12 Punkte)

2. Finden und demonstrieren Sie mit der Applikation aus Aufgabenteil 1. einen Eingabewert, der eine Folge erzeugt, die mit einem Zyklus beginnt und einen Eingabewert, der eine Folge erzeugt, die einen Zyklus enthält, aber nicht damit beginnt.  
(4 Punkte)
3. Finden Sie eine Folge, die mit einem Zyklus der Länge 2 beginnt und eine Folge, die einen Zyklus der Länge 2 enthält, aber nicht damit beginnt.  
(4 Punkte)
4. Mit dem *Hase-Igel-Algorithmus* kann ein Zyklus in einer Folge mit einfachen rekursiven Bildungsgesetz gefunden werden.
  - Die Folge wird mit Hase und Igel beginnend beim ersten Folgenglied mit unterschiedlichen Schrittweiten durchlaufen.
  - Bei jeder Iteration wird der Hase zwei und der Igel ein Folgenglied weiterbewegt.
  - Begegnen sich der Hase und der Igel, d.h. beide haben denselben Wert, enthält die Folge einen Zyklus.

Benutzen Sie den *Hase-Igel-Algorithmus*, um alle Startwerte für die Applikation aus Aufgabenteil 1 zu finden, die Folgen erzeugen, welche einen Zyklus der Länge größer 1 enthalten.

- Berechnen Sie für jeden Startwert mit der *Mittelquadratmethode* solange Werte für Hase und Igel bis ein passender Zyklus gefunden wird.

Bemerkung.

Da nur 100 verschiedene zweistellige Zahlen zur Verfügung stehen, muss innerhalb der ersten 101 Listenelemente mindestens eine Zahl doppelt vorkommen, d.h. ein Zyklus gefunden werden.

- Ein Zyklus ist länger als 1, wenn sich der Wert des Igels und der auf den Igel folgende Wert unterscheiden.
- Kennzeichnen Sie alle wichtigen Schritte, z.B. das Aktualisieren von Hase und Igel, durch Kommentare.

(15 Punkte)