

Übungsblatt 04

E-Learning

Absolvieren Sie die Tests bis Di., 20.11., 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 – 22 Punkte

Wahrheitswerte und Boolesche Gesetze

Lesen Sie **bevor Sie den Test starten** das Kapitel 7.5 Aussagenlogik in Skript.

Absolvieren Sie den Test *Informatik I - ILIAS 04 Teil 1*.

(22 Punkte)

Hinweis

Wenn Sie den Test einmal vollständig durchlaufen haben kommen Sie auf die Seite *Testergebnisse*. Starten Sie den Test erneut aus Stud.IP, ist jetzt auch eine Schaltfläche *Testergebnisse anzeigen* vorhanden, die auf diese Seite führt.

Auf der Seite *Testergebnisse* können Sie sich unter *Übersicht der Testdurchläufe* zu jedem Testdurchlauf *Details anzeigen* lassen.

In der Auflistung der Aufgaben führt der Titel einer Aufgabe zu einer **Musterlösung** für die jeweilige Aufgabe.

ILIAS 4-Minuten-Aufgaben – 12 Punkte

Absolvieren sie die Tests *Informatik I - ILIAS 04 Teil 2*, *Teil 3* und *Teil 4*.

(12 Punkte)

LON-CAPA – 18 Punkte

Zahlensysteme

Absolvieren Sie im Test *Informatik I - LON-CAPA* die *Übung 04*.

(18 Punkte)

Übung

Abgabe bis Di., 20.11., 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.

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 – 13 Punkte

Codierung von Sprachen

1. In den 50ziger Jahren werden bei Ausgrabungen Listen auf Tontafeln gefunden. Die dabei verwendeten Glyphen (Schriftzeichen) sind bis dato nicht bekannt. Die Listen sollen mit einem Computersysteme verarbeitet werden, dazu müssen die Glyphen binär codiert werden.

Aufgrund der Beschränkungen der damaligen Computersysteme, wird eine möglichst effiziente Codierung gesucht.

- a) Die Schrift hat 22 Glyphen (Schriftzeichen).

Wieviel Bits sind erforderlich um die Glyphen einzeln binär zu codieren?
(1 Punkt)

- b) Bei der Untersuchung der Listen fällt dem Archäologen Dr. Henry Jones Jr. auf, dass alle Einträge eine gerade Anzahl von Glyphen haben. Deshalb schlägt er vor nicht einzelne Glyphen, sondern Paare von Glyphen binär zu codieren.

Wieviele Bits sind erforderlich für die Glyphenpaare?
(1 Punkt)

- c) Erklären Sie, warum die Codierung der Listeneinträge mit der Methode aus Aufgabenteil 1b) tatsächlich eine Verbesserung gegenüber der Methode aus Aufgabenteil 1a) ist.

(2 Punkte)

2. Die Verarbeitung der Listen mit dem Computersystem hat Erfolg, es kann nachgewiesen werden, dass es sich um Königslisten einer untergegangenen Hochkultur handelt.

Einige Jahre nach Codierung der Königslisten werden weitere Texte gefunden. Diesmal finden, zusätzliche zu den 22 bekannten Glyphen, weitere 4 Zeichen Verwendung.

Der erneut hinzugezogene Dr. Jones macht folgende Entdeckungen.

- Die 4 neuen Zeichen sind Zahlzeichen.
- Eine der 22 Glyphen, die ' -Glyphe, hat eine spezielle Bedeutung als Abschluss-Glyphe, das war bei den Königslisten noch nicht aufgefallen. Alle Worte oder Zahlen enden mit der ' -Glyphe, die ausschließlich als letztes Zeichen eines Wortes oder einer Zahl vorkommt.
- Worte bestehen nur aus Glyphen. Zahlen bestehen, mit Ausnahme des letzten Zeichens (' -Glyphe), nur aus Zahlzeichen.
- Jedes Wort besteht, inklusive der ' -Glyphe, aus einer geraden Anzahl von Glyphen.
- Die Anzahl der Zahlzeichen in einer Zahl unterliegt keiner erkennbaren Gesetzmäßigkeit.
- Außer der ' -Glyphe gibt es keine weiteren Wort-/Zahltrenner, Satzzeichen oder Ähnliches.

Die bei den Königslisten erfolgreich verwendete Codierung von Glyphenpaaren soll auch bei den neu entdeckten Texten angewendet werden.

- a) Wieviel Bits sind erforderlich um alle Zeichen einzeln binär zu codieren?
(1 Punkt)
- b) Wieviele Bits sind erforderlich wenn man Glyphen paarweise und die restlichen Zeichen einzeln binär codiert?
(1 Punkt)

Hinweis. Vergessen Sie nicht, das Zahlen mit der ' -Glyphe enden.

- c) Zeigen Sie, das die Codierung eines Textes mit der Methode aus Aufgabenteil 2b) nicht in jedem Fall besser ist, als die Codierung mit der Methode aus Aufgabenteil 2a).
(2 Punkte)

Hinweis. Geben Sie ein Gegenbeispiel an.

- d) Verbessern Sie die Methode aus Aufgabenteil 2b), indem Sie keine einzelnen Zeichen mehr codieren, sondern Paare und Tripel.
(2 Punkte)
- e) Erklären Sie, warum die Codierung von Texten mit mindestens 2 Zeichen mit der Methode aus Aufgabenteil 2d) tatsächlich eine Verbesserung gegenüber der Methode aus Aufgabenteil 2a) ist?
(3 Punkte)

Praktische Übung

Abgabe der Prüfsumme bis Di., 20.11., 23:55 Uhr.

Testat von Mi., 28.11., 8-10 Uhr bis Mo., 03.12., 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 04 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 – 10 Punkte

Felder

Für n positive Zahlen, x_1, x_2, \dots, x_n mit $x_i > 0$ für $i = 1, \dots, n$, kann das arithmetische Mittel (der Durchschnitt) a und das geometrische Mittel b wie folgt berechnet werden.

$$a = \frac{1}{n} \sum_{i=1}^n x_i \qquad b = \sqrt[n]{\prod_{i=1}^n x_i}$$

Beispiele

- Für beliebiges n und $x_i = 1$ mit $i = 1, \dots, n$ sind arithmetisches Mittel und geometrische Mittel gleich 1.
- Für $x_1 = 1, x_2 = 9$ ist das arithmetische Mittel 5 und das geometrische Mittel 3.

Schreiben Sie eine Applikation, die stichprobenartig überprüft, ob das arithmetische Mittel größer gleich dem geometrischen Mittel ist, wie in den betrachteten Beispielen.

- Ein `int`-Wert n wird von der Kommandozeile übernommen. Es wird geprüft, ob $n > 1$ gilt, wenn nicht wird eine Fehlermeldung ausgegeben und die Applikation beendet.
- Ein `int`-Feld der Länge n wird erzeugt und mit Zufallszahlen aus dem Bereich $1, \dots, n$ gefüllt. Das Feld wird ausgegeben.
- Das arithmetische und das geometrische Mittel werden berechnet, dabei wird nur eine Schleife verwendet. Arithmetisches und geometrisches Mittel werden ausgegeben.

Hinweis

Die Wurzel kann man mit `Math.pow` berechnen, z.B. berechnet `Math.pow(z, 1.0/n)` die n -te Wurzel aus z .

- Gilt arithmetisches Mittel größer gleich geometrischen Mittel wird OK, sonst WIDERLEGT ausgegeben.

Hinweis. Siehe `class RandomInt` und `class PrimeSieve` aus Kapitel 2 des Skripts.
(10 Punkte)

Bemerkung

Man kann beweisen, dass für beliebige n positive Zahlen das arithmetische Mittel größer gleich dem geometrischen Mittel ist.

Aufgabe 2 – 25 Punkte

StringToFloat

Rekapitulieren Sie Kapitel 2.1.2 im Skript und benutzen Sie zu Klärung offener Fragen den *IEEE Standard for Binary Floating-Point Arithmetic* (Stud.IP→Informatik I→Dateien→Übungsblätter→Daten).

Schreiben Sie eine Applikation, die eine auf der Kommandozeile übergebene Zeichenkette der Länge 32, die nur die Zeichen '0' und '1' enthält, verarbeitet.

Zeichenketten im falschen Format führen zu einer Fehlermeldung und dem Programmende. Zeichenketten im richtigen Format werden als binäre Codierung einer Gleitkommazahl mit einfacher Genauigkeit nach IEEE-Standard 754 interpretiert.

- NaN, +0, -0, ∞ und $-\infty$ werden erkannt und es wird NaN, +0, -0, `NEGATIVE_INFINITY` oder `POSITIVE_INFINITY` ausgegeben.
- Es wird ermittelt, ob es sich um eine Gleitkommazahl in normalisierter oder denormalisierter Darstellung handelt.

- Es werden `int`-Werte für das Vorzeichen $s \in \{0, 1\}$ und den verschobenen Exponenten e , sowie `float`-Werte für die reduzierte Mantisse f und die resultierende Gleitkommazahl v , nach einer der folgenden Formeln ermittelt.

$$v = (-1)^s \cdot 2^{e-127} \cdot (1.f) \quad (\text{normalisiert})$$

$$v = (-1)^s \cdot 2^{-126} \cdot (0.f) \quad (\text{denormalisiert})$$

Die Werte s , e , f und v werden ausgegeben.

Benutzen Sie folgende Klasse als Grundlage für Ihre Implementierung.

```
class StringToFloat {
    public static void main( String args[] ) {
        String str = args[0];
        for (int i = 0; i < str.length(); i++)
            System.out.print(str.charAt(i));
        System.out.println();
    }
}
```

(25 Punkte)