



Name:

Vorname:

Geburtsdatum:

Matrikelnummer:

Platz:

Nachklausur
Informatische Werkzeuge in den
Geistes- und Sozialwissenschaften 1

11. April 2024

	To be used for grading, do not write here											
prob.	1.1	1.2	1.3	1.4	2.1	2.2	3.1	3.2	4.1	4.2	Sum	grade
total	4	4	4	4	6	9	6	5	6	12	60	
reached												



Organisatorisches

Bitte lesen die folgenden Anweisungen genau und bestätigen Sie diese mit Ihrer Unterschrift.

1. Bitte legen Sie Ihren Studierendenausweis und einen Lichtbildausweis zur Personenkontrolle bereit!
2. Sie können 60 Punkte erreichen, wenn Sie alle Aufgaben vollständig lösen. Allerdings zählen 55 Punkte bereits als volle Punktzahl, d.h. 5 Punkte sind Bonuspunkte.
3. Es sind keine Hilfsmittel erlaubt außer eines handgeschriebenen "Spickzettels" von 1 Seite A4 einseitig.
4. Die Bearbeitungszeit beträgt genau 60 min.
5. Schreiben Sie die Lösungen direkt auf die ausgeteilten Aufgabenblätter. Andere Blätter werden nicht bewertet.
6. Wenn Sie die Prüfung aus gesundheitlichen Gründen abbrechen müssen, so muss Ihre Prüfungsunfähigkeit durch eine Untersuchung in der Universitätsklinik nachgewiesen werden. Melden Sie sich in jedem Fall bei der Aufsicht und lassen Sie sich das entsprechende Formular aushändigen.
7. Überprüfen Sie Ihr Exemplar der Klausur auf Vollständigkeit (11 Seiten inklusive Deckblatt und Hinweise) und einwandfreies Druckbild! **Vergessen Sie nicht, auf dem Deckblatt die Angaben zur Person einzutragen und diese Erklärung zu unterschreiben!**

Erklärung: Durch meine Unterschrift bestätige ich den Empfang der vollständigen Klausurunterlagen und die Kenntnisnahme der obigen Informationen.

Erlangen, 11. April 2024

.....
(Unterschrift)

Bitte beachten Sie die folgenden Regeln, um keine Punkte zu verlieren:

- Wenn Sie eine Antwort auf einer anderen Seite fortsetzen, geben Sie bitte die Nummer der Aufgabe auf der neuen Seite mit an und verweisen Sie auf der alten Seite auf die neue.
- Begründen Sie Ihre Aussagen, wenn angebracht (wir würden gerne Teilpunkte für unvollständige Antworten geben). Wenn nicht explizit darum gebeten, antworten Sie möglichst nicht einfach mit „Ja“, „Nein“ oder „42“.

1 Grundlagen und Verständnis



Aufgabe 1.1 (Dateipfade)

4 Punkte

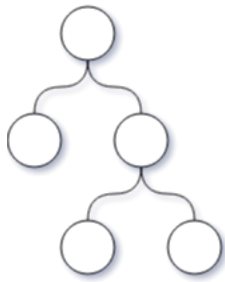
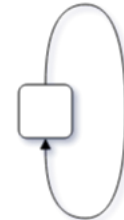
Was ist ein *Dateipfad*? Wie unterscheidet sich dieser von einem *file name* Dateinamen? Was bedeuten in diesem Kontext *relativ* und *absolut*?



Aufgabe 1.2 (Bäume)

4 Punkte

Welche der folgenden Graphen zeigen korrekte Bäume im mathematischen Sinne? Für nicht-Bäume begründen Sie bitte Ihre Entscheidung kurz (!).

**A****B****C****D**

**Aufgabe 1.3 (17 in verschiedenen Zahlensystemen)**

4 Punkte

In Stellenwert-Zahlensystemen (positional number systems) hängt der Wert einer Zahl mit mehr als einer Stelle davon ab, in welchem System wir uns genau befinden.

Geben Sie jeweils den Wert der Zahl 17_{10} in folgenden Zahlensystemen an:

1. Hexadezimalsystem (Basis 16)
2. Dezimalsystem (Basis 10)
3. Binärsystem (Basis 2)
4. Unärsystem (Basis 1)

**Aufgabe 1.4 (HTTP Requests)**

4 Punkte

Nennen Sie die beiden wichtigen strukturellen Bestandteile einer HTTP POST-Anfrage. Welchen dieser Teile hat eine HTTP GET-Anfrage in der Regel *nicht*? Angenommen, dass mit dieser POST-Anfrage eine Bilddatei auf einen Server hochgeladen werden soll, in welcher der beiden Komponenten, würden diese Daten übertragen?

2 Reguläre Ausdrücke



Aufgabe 2.1 (Matchen Gegebener Regexes)

6 Punkte

Geben Sie einen String an, der gegen alle folgende regulären Ausdrücke matcht.

- `..+apyb.+`
- `^[Cc].{7}$`
- `.a...a.a`
- `.*ara.*`

**Aufgabe 2.2 (RegEx für Go-Notation)**

9 Punkte

Das chinesische Strategiespiel *Go* wird (normalerweise) auf einem Brett mit 19x19 Feldern für die Figuren (gen. *Steine*) gespielt. Diese Felder sind auf der einen Seite mit den Buchstaben A bis T (der Buchstabe I wird ausgelassen), auf der anderen Seite mit den Zahlen 1 bis 19 gekennzeichnet.

Um den Verlauf eines Spiels später nachvollziehen zu können werden die einzelnen Züge (ähnlich wie auch beim Schach) notiert. Eine Notation besteht dabei aus einer Zahl (die theoretisch beliebig groß sein kann) für den Zug (Start bei 0) gefolgt von einem Punkt, einem beliebigen (!) Whitespace-Zeichen und einem gültigen Feldnamen. Geben Sie einen regulären Ausdruck für solche Notationen an.

Hinweis: Hier sind ein paar gültige Notationen für Züge in Go:

- 01. A1
- 8. T19
- 9999999. N10

Die folgenden Notationen hingegen wären *nicht* gültig:

- 1. A100
 - 2. I10
 - 17. Z0
-

3 Digitale Dokumente



Aufgabe 3.1 (XPath)

6 Punkte

Beschreiben Sie für jeden der unten angegebenen XPaths, welche Elemente in einem XML-Dokument genau damit erreicht werden können.



1. `/city/library/book[@lang="de"] [314]`



2. `//div[@id="hero"]//img`

**Aufgabe 3.2 (Mentales CSS)**

5 Punkte

Gegeben ist der Quelltext der folgenden HTML-Seite:

```
<!DOCTYPE html>
<html lang="en">
  <style>
    body { background-color: grey; }
    .foo { background-color: red; }
    .bar { background-color: black; }
    #baz { background-color: green; }
    .foo.bar { background-color: yellow; }
    .qux { background-color: pink; }
    .xuq { background-color: white; }
    .bar { background-color: blue; }
  </style>
  <body>
    <ul>
      <li class="foo">One</li>
      <li class="bar">Two</li>
      <li class="baz">Three</li>
      <li id="qux">Four</li>
      <li class="bar_foo">Five</li>
    </ul>
  </body>
</html>
```

Beschreiben Sie, welche Elemente auf der fertig dargestellten Seite zu sehen sind und welche Hintergrundfarbe sie haben.

4 Programmieren in Python



Aufgabe 4.1 (Primzahlen)

6 Punkte

Schreiben Sie eine python-Funktion namens `isPrime`, die einen Integer entgegen nimmt und einen Wahrheitswert (Boolean) zurück gibt, ob diese Zahl eine Primzahl ist.

Hinweis: Eine Primzahl ist nur durch sich selbst und 1 ganzzahlig teilbar und hat keine anderen Faktoren. Die Zahl 1 ist keine Primzahl. Die ersten Primzahlen sind also 2, 3, 5, 7, 11 ...



Aufgabe 4.2 (Bäume mit lxml)

12 Punkte

Setzen Sie sich genau mit folgendem Schnipsel python-Code auseinander. Es wird die in der Vorlesung besprochene lxml-Bibliothek verwendet um iterativ (d.h. in mehreren Durchläufen) eine Baum-Datenstruktur aufzubauen, in der jeder Knoten in seinem `.text`-Attribut einen String enthält, der einer Zahl entspricht.

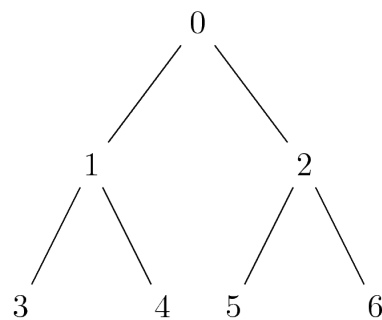
```
from lxml import etree

# Lege die Wurzel des Baumes an
root = etree.Element("node")
root.text = "0"
counter = 1

# Tue genau zwei Mal folgendes:
for _ in range(0,2):
    # Fuer "root" und alle seine Nachfahren tue folgendes:
    for element in root.iter():
        # Wenn der Knoten keine Kinder hat...
        if len(element) == 0:
            # ... fuege diesem Knoten (element) zwei Kinder hinzu ...
            fst_child = etree.SubElement(element, "node")
            snd_child = etree.SubElement(element, "node")
            # ... und gebe allen neuen Kindern den aktuellen counter als text.
            for child in element:
                child.text = str(counter)
                counter = counter + 1
```

Wird der obige Code ausgeführt, so ist es möglich, den dort generierten Baum mit der Wurzel `root` wie folgt darzustellen (einmal in XML-Syntax und einmal als Grafik):

```
<node>
  0
  <node>
    1
    <node>3</node>
    <node>4</node>
  </node>
  <node>
    2
    <node>5</node>
    <node>6</node>
  </node>
</node>
```



Eine bildliche Darstellung des links beschriebenen Baumes.

Schreiben Sie nun eine python-Funktion `renameNodes`, die den Wurzelknoten eines solchen Baums als Argument bekommt. Die Funktion soll alle Knoten in dem Baum besuchen und ggf. deren `.text`-Attribut ersetzen. Dabei soll der neue Wert des Attributs "prim!" sein, wenn der ursprüngliche Wert des Attributs eine Primzahl darstellt. Andernfalls soll der Wert *nicht* ersetzt werden.

Hinweis: Sie können davon ausgehen, dass Ihnen eine korrekt implementierte Funktion `isPrime` wie aus der vorherigen Aufgabe zur Verfügung steht (auch wenn Sie diese Aufgabe nicht bearbeitet haben). Alle sonstigen Methoden und Funktionen, die Sie für die Beantwortung dieser Aufgabe benötigen, werden auch im obigen Codeschnipsel verwendet und werden deshalb hier nicht noch einmal aufgezählt.

Bedenken Sie, dass der Wert des `.text`-Attributs ein String ist, auch wenn er nur Ziffern enthält.



This page was intentionally left blank for extra space



This page was intentionally left blank for extra space