Name: Vorname:

Geburtsdatum: Matrikelnummer:

Platz:

Klausur
Informatische Werkzeuge in den Geistes- und Sozialwissenschaften 1

16. Februar 2023

	To be used for grading, do not write here										
prob.	1.1	1.2	1.3	2.1	2.2	3.1	3.2	4.1	4.2	Sum	grade
total	4	4	4	6	8	8	10	6	10	60	
reached											

Die „Lösungen“ der Aufgaben in diesem Dokument sollen den Studierenden als Anfangspunkt für die Beantwortung der Aufgaben dienen. Trotz aller Bemühungen kann es zu Unvollständigkeiten oder sogar Fehlern in den „Lösungen“ kommen. Da die Korrektur und Benotung der gestellten Aufgaben niemals lediglich auf einen „Vergleich mit der Musterlösung“ hinausläuft, ist dies auch nicht so schlimm. In jedem Fall sollten die Studierenden die Lösungen nachvollziehen und im Prinzip mittels des Lehrstoffs selbst verifizieren können.

Sollten Sie „Lösungen“ finden, die Sie nicht verstehen oder sogar für fehlerhaft halten diskutieren Sie diese am besten mit den Tutor*innen oder auf dem Kursforum und benachrichtigen Sie die Lehrenden; wir werden sie dann gegebenenfalls baldigst korrigieren.

1 Grundlagen und Verständnis

Aufgabe 1.1 (Stellenwertsysteme) 4 Pkt
Menschen benutzen verschiedene Zahlensysteme. Manche davon sind *Stellenwertsysteme* („Positional Number Systems“). Erklären sie kurz (!), was ein Stellenwertsystem ausmacht. Nennen Sie außerdem zwei Zahlensysteme, die Stellenwertsysteme sind und ein Zahlensystem, welches keins ist. 4 min

---

Solution: In einem Stellenwertsystem ist der Beitrag einer einzelnen Ziffer zur Gesamtzahl abhängig von der Stelle (i.S.v. Position), an der sich die Ziffer befindet.

Korrekte Antworten für Stellenwertsysteme sind z.B. das Binär-, Oktal-, Dezimal- oder Hexadezimalsystem. Korrekte Antworten für nicht-Stellenwertsysteme sind z.B. das römische oder ägyptische Zahlensystem.

Korrektur: Das Unärsystem ist zwar *technisch* gesehen ein Stellenwertsystem, da aber $1^0 = 1^1 = 1^2 = \cdots = 1$, kann es auch als ein Additionssystem (und damit nicht-Stellenwertsystem) verstanden werden.

Aufgabe 1.2 (URI-Encoding) 4 Pkt
Erklären Sie kurz (!), was beim *URI-Encoding* vor sich geht. Was genau wird encodiert? Warum/wann ist das nötig? 4 min

---

Solution: Beim URI-Encoding werden Texte mit Sonderzeichen (jenseits von ASCII) in ASCII-kompatible Texte umgewandelt (um Fehler bei der Kommunikation zwischen Computern, z.B. im Internet zu verhindern). Für jedes Zeichen passiert folgendes:

- Sonderzeichen wird als UTF-8 (Unicode) dargestellt.
- Jedes Byte dieses Formats wird zu drei ASCII-Zeichen konvertiert.
  - Das erste Zeichen ist immer “%”.
  - Die letzten zwei Zeichen sind die Hexadezimalrepräsentation des Byte-Werts.

Korrektur: Eine Punkte für eine korrekte, aber allgemeine Beschreibung, zwei für eine Konkrete (die Bytes und das Prozent-Zeichen erwähnt). Einen Punkt für korrekt Beantwortung der konkreten Fragen.

Aufgabe 1.3 (Bäume) 4 Pkt
Was ist ein *Baum* im Kontext von Informatik? Welche Bestandteile haben Bäume? Erklären Sie außerdem kurz die Begriffe *Wurzel* und *Blatt*. 4 min

Solution: Ein Baum ist eine Datenstruktur, die die hierarchische Anordnung von Objekten (genannt: Knoten) ermöglicht. Jedem Knoten K kann dabei eine Menge von Knoten niedriger in der Hierarchie zugeordnet werden. Diese Menge nennen wir die Kinder von K, wobei K deren Elternteil ist.

In einem Baum darf kein Knoten mehr als ein Elternteil haben. Die Eltern-Kind-Relation darf außerdem keine Zykel enthalten.

Der Knoten ohne Eltern (pro Baum gibt es nur einen) wird Wurzel genannt. Knoten ohne Kinder sind Blätter.

2 Reguläre Ausdrücke

Aufgabe 2.1 (Regulärer Ausdruck für Uhrzeit) 6 Pkt
Geben Sie einen regulären Ausdruck an, der auf textuelle Angaben der Uhrzeit im Format `hh:mm:ss` (24 Stunden-Format mit Minuten und Sekunden) matcht. 6 min

Hier sind ein paar Uhrzeiten im korrekten Format:

```
09:30:59
13:12:00
22:49:22
```

Diese hier hingegen sind inkorrekt und sollen *nicht* gematcht werden.

```
12:04:11 AM
32:80:99
4:1:7
```

Falls Sie Gruppierungen verwenden, ist es egal ob diese *capturing* oder *non-capturing* sind.

Solution: Ein solcher Ausdruck wäre zum Beispiel:

```
(?:[01][0-9]|2[0-3]):[0-5][0-9]:[0-5][0-9]
```

Aufgabe 2.2 (Regulärer Ausdruck für DOIs) 8 Pkt
Um auf digitale Objekte (wie Journals, Berichte, Paper, Artikel, ...) verlässlich verweisen zu können, wurde von der International Organization for Standardization (ISO) im Jahr 2012 ein Standard für *Digital Object Identifiers* (**DOI**) veröffentlicht. 8 min

- Ein **DOI** besteht aus einem Präfix und einem Suffix, die von einem „/“ getrennt werden.
- Das Präfix beginnt (für unsere Zwecke) immer mit der Zeichenkette „`10.`“ gefolgt von einer ganzen Zahl ≥ 1000.
- Das Suffix ist beliebig und unterliegt keinerlei Einschränkungen. In der Praxis ist es oft numerisch, der Standard erlaubt allerdings jede (nichtleere) Aneinanderreihung von Zeichen.

In normalen Texten ohne Hyperlinks wird ein **DOI** oft mit der Zeichenkette „`doi:`“ eingeführt, wie auch in den folgenden Beispielen:

```
doi:10.1000/182
doi:10.20347/WIAS.PREPRINT.2431
doi:10.1016/j.disc.2012.08.018
doi:10.1007/978-3-662-44199-2_5
doi:10.1007/s10817-012-9271-4
```

Geben Sie einen regulären Ausdruck an, der auf textuelle **DOI**-Ausdrücke wie oben beschrieben (also inklusive „`doi:`“) matcht. Falls Sie Gruppierungen verwenden, ist es egal ob diese *capturing* oder *non-capturing* sind.

Solution: Ein solcher Ausdruck wäre zum Beispiel:

```
doi:10\.[1-9]\d{3}\d*/.+
```

---

3 Digitale Dokumente

Aufgabe 3.1 (XPath) 8 Pkt
8 min

1. Im Kontext von `XML`-Dokumenten, was versteht man unter einem *XPath*?
2. Beschreiben Sie außerdem für jeden der unten angegebenen XPaths, was genau damit erreicht werden kann.
   (a) `//*[@price<42]`
   (b) `//img[@alt='Logo']`

Solution: Die *XML Path Language* ist eine Abfragesprache für XML-Dokumente, ähnlich wie reguläre Ausdrücke für unstrukturierten Text. Ein XPath-Ausdruck adressiert eine bestimmte Menge an Knoten eines XML-Dokuments, das dabei als Baum betrachtet wird.

1. Dieser *XPath* beschreibt die Menge aller Knoten im Dokument, die ein Preis-Attribut von weniger als 42 haben.
2. Dieser XPath wählt alle die `img`-Knoten im Dokument aus, deren `alt`-Attribut (genutzt für Alternativtext / Bildbeschreibung) den Wert “Logo” hat.

Korrektur:

- 4 Punkte für korrekte Beschreibung des Konzepts
- Je 2 Punkte für korrekte Beschreibung eines XPaths

Aufgabe 3.2 (HTML Seite mit CSS) 10 Pkt
Schreiben Sie den Quellcode für eine gültige *HTML*-Seite. 10 min

1. Diese Seite sollte im `body`-Teil mindestens drei *unterschiedliche* Elemente zeigen, mit denen Nutzer*Innen in irgendeiner Form interagieren können.

2. Die Seite sollte weiterhin im `head`-Bereich in angemessener Umgebung mindestens drei CSS-Regeln beinhalten. Diese Regeln sollen die Elemente im `body` visuell verändern. Achten Sie bitte dabei darauf, dass sich keine Regelelemente doppeln. Gegebene Selektoren und veränderte Attribute sollten also höchstens einmal auftauchen. Ästhetische Bedenken spielen in dieser Aufgabe *keine* Rolle.

3. Zuletzt soll außerdem irgendwo im Quellcode ein gültiger HTML-Kommentar vorkommen.

Solution: Hier ist eine von vielen Möglichkeiten, diese Aufgabe zu lösen:

```
<!DOCTYPE html>
<html>
    <head>
        <style type="text/css">
            form { width: 150px; }
            body { background-color: #0ACAB0; }
            input { margin: 2px 0 }
        </style>
    </head>
    <body>
        <!-- Form with two input elements and a submit button -->
        <form>
            <input type="text" placeholder="Username">
            <br/>
            <input type="password" placeholder="Password">
            <hr/>
            <input type="submit" value="Submit">
        </form>
    </body>
</html>
```

Korrektur:

- Drei Punkte für ein prinzipiell korrektes HTML-Gerüst.
- Einen Punkt für einen korrekten HTML-Kommentar.
- Je einen Punkt für ein gültiges Interaktionselement.
- Je einen Punkt für eine gültige CSS-Regel.

---

4 Programmieren in Python

Aufgabe 4.1 (Python-Verständnisfrage) 6 Pkt
6 min

Machen Sie sich mit den in Abbildung 1 definierten Funktionen `a(s1,s2)` und `m(f1,f2)` vertraut. Welche gängigen Funktionen sind hier verschleiert implementiert worden? Was ist der Rückgabewert von `m(4,5)`?

```
# Slightly obfuscated code
def a(s1, s2):
    x = 0

    for _ in range(s1):
        x = x + 1

    for _ in range(s2):
        x = x + 1

    return x

def m(f1, f2):
    y = 0
    z = 0

    while z < f1:
        y = a(y, f2)
        z = z + 1

    return y
```

?figurename? 1: Welche bekannten Funktionen sind hier zu sehen?

Solution:

- `a(s1,s2)` *addiert* die beiden Argumente (Summanden) und gibt das Ergebnis zurück.
- `m(f1,f2)` *multipliziert* die beiden Argumente (Faktoren) und gibt das Ergebnis zurück.
- `m(4,5)` $= 4 \cdot 5 = 20$

Aufgabe 4.2 (Bottle Route für Reguläre Ausdrücke) 10 Pkt
10 min

Schreiben Sie eine Route `regex()` für einen `bottle`-Server, die zwei Argumente (Strings) über die URL annimmt: einen regulären Ausdruck und einen Text. Es soll getestet werden ob und wenn ja wie der RegEx auf den Text matcht.

Die Route soll konkret folgendes Verhalten zeigen:

- Matcht der RegEx an keiner Stelle im String, so soll ‘‘`Kein Match!`’’ zurück gegeben werden.
- Matcht der RegEx den gesamten String von Anfang bis Ende, so soll ‘‘`Komplettes Match!`’’ zurück gegeben werden.

- Matcht der RegEx nur partiell, so soll angegeben werden, an wie vielen Stellen dies passiert (z.B. ‘‘`Genau 4 Matches!`’’). Nur die Anzahl ist wichtig, nichts weiter.

Sie können für diese Aufgabe davon ausgehen, dass alle benötigten `import`-Statements bereits vorhanden sind und das alle Strings wie getippt an die Route weiter gegeben werden.

Solution: Eine mögliche Lösung ist die folgende:

```
@route('/regex/<needle>/<haystack>')
def regex(needle, haystack):
    match_result = re.findall(needle, haystack)

    if len(match_result) == 0:
        return "Kein␣Match!"
    else:
        # Erstes Ergebnis ist relevant:
        first = match_result[0]

        if len(first) == len(haystack):
            return "Komplettes␣Match!"
        else:
            amount = len(match_results)
            return f"Genau␣{amount}␣Matches!"
```
