

## Assignment 4

### Abgabe bis 8. Woche

Je nach Halbgruppe 24./25. April,

### Achtung!

Jeder Studierende muss in der 8. Woche eine gelöste Aufgabe abgeben.

### Anforderungen:

- Jede Aufgabe wird in einer eigenständigen Datei (.sh) gelöst .
- Jede Halbgruppe lädt die Lösungen in das ihrer Halbgruppe entsprechende Assignment.
- Die Dateien werden nach folgendem Format benannt und hochgeladen:

**Name\_Vorname\_Gruppe\_X.sh**

wobei **X** die Nummer der Aufgabe ist, die Sie lösen müssen.

- Wer andere als die zugewiesenen Aufgaben löst, wird mit Note 0 bewertet.

**Zuordnung der Aufgaben:** Die zur Lösung vorgeschlagenen Aufgaben werden in der Laborstunde der 6. Woche verteilt.

## Aufgabe 1

Der Name einer Datei mit der Erweiterung ".c" wird als Parameter in der Befehlszeile angegeben. Bestimmen Sie die Anzahl der Aufrufe für jede im Programm definierte "int"-Funktion (wie viele Male ist die "int" -Funktion im Programm ausgeführt). Das Ergebnis wird in der Standardausgabe wie folgt angezeigt:

"Name\_der\_Funktion Anzahl\_der\_Aufrufe"

### Beispiel für die Standardausgabe:

summe 3

differenz 2

## Aufgabe 2

Der Name einer Textdatei wird über die Befehlszeile bereitgestellt. Erstellen Sie ein Ordner mit dem Namen "Wörterbuch", das 26 Dateien enthält, die nach folgendem Format organisiert sind:

- die erste Datei wird "A" genannt. Es enthält alle Wörter in der als Parameter bereitgestellten Textdatei, die mit dem Buchstaben 'a' (oder 'A') beginnen. Die Wörter werden alphabetisch sortiert und einzeln in der Datei (=jedes Wort auf einer Zeile) angezeigt.
- die zweite Datei wird "B" genannt. Es erfüllt alle Spezifikationen der ersten Datei.
- Die restlichen Dateien erfüllen alle Spezifikationen der ersten Datei und repräsentieren den Rest der Buchstaben des englischen Alphabets.

Bei der Standardausgabe wird nichts angezeigt. Alle 26 Wörterbuchdateien werden erstellt, auch wenn einige davon leer sind.

### Aufgabe 3

Die Datei "studenten.csv" enthält die Namen der Studenten des 1. Jahres und ihre allgemeinen Mittelnoten aus dem ersten Semester. Die Liste der Studierenden ist absteigend, nach ihrem allgemeinen Durchschnitt, sortiert. Die Datei "notenSommer.csv" enthält die Namen der Studierenden in alphabetischer Reihenfolge und 5 weitere Noten, die die Noten des zweiten Semesters darstellen. Aktualisieren Sie die Datei „studenten.csv“, so dass sie die Studenten-Rangliste am Ende des 1. Jahres enthält. Die endgültige Datei enthält die Namen der Studenten sowie den allgemeinen Durchschnitt des gesamten Universitäts-Jahres. Die Daten in den beiden Dateien werden durch Kommas abgetrennt. Bei der Standardausgabe wird nichts angezeigt.

#### Beispiel:

| (studenten.csv)     | (notenSommer.csv)           |
|---------------------|-----------------------------|
| popescu ion,10      | ifrim marius,9,7,10,10,9    |
| tomulescu camelia,9 | popescu ion,10,8,9,10,9     |
| ifrim marius,8      | nulescu camelia,9,10,10,9,8 |

### Aufgabe 4

Ein Ordner und eine Liste von Dateien werden als Parameter auf der Befehlszeile bereitgestellt. Der Ordner ist immer der erste Parameter. Zeigen Sie für jede Datei alle Unterordnern an, in denen diese vorkommen. Das Ergebnis wird in der Standardausgabe im folgenden Format angezeigt: Alle Informationen zu einer bestimmten Datei werden in derselben Zeile angezeigt. Die Datei gefolgt von ":" wird zuerst angezeigt, und dann werden die Unterordnern angezeigt, zu denen sie gehört (durch Leerzeichen getrennt).

#### Beispiel für die Standardausgabe:

```
fis1:subdir11 subdir12 subdir13
fis2:subdir22 subdir22
```

## Aufgabe 5

Die Namen zweier Ordner werden in der Befehlszeile angegeben. Verschieben Sie den vom zweiten Ordner bezeichneten Zweig als Unterzweig des ersten. Für jede Datei im zweiten Ordner wird die Erweiterung in „.eins“ geändert. Bei der Standardausgabe wird nichts angezeigt.

## Aufgabe 6

Der Name eines Ordners wird als Parameter in der Befehlszeile angegeben. Zeigen Sie die Liste aller Dateien und Unterordnern an, die in der Baumstruktur des angegebenen Ordners erscheinen. Außerdem wird für jeden die Anzahl der Auftritte in der Struktur angezeigt. Das Ergebnis wird alphabetisch sortiert bei der Standardausgabe angezeigt. Jede Datei / jeder Ordner wird in einer Zeile angezeigt, gefolgt von der Anzahl der Vorkommen. Die Daten werden durch Leerzeichen abgetrennt.

### Beispiel für die Standardausgabe:

```
director1 3
fis1 2
fis2 1
```

## Aufgabe 7

Der Name einer Textdatei wird über die Befehlszeile bereitgestellt. Erstellen Sie ein Ordner mit dem Namen "Wörterbuch", das 10 Dateien enthält, die nach folgendem Format organisiert sind:

- die erste Datei wird "0" genannt. Es enthält alle Wörter in der als Parameter bereitgestellten Textdatei, die mit dem Ziffer '0' beginnen. Die Wörter werden alphabetisch sortiert und einzeln in der Datei (=jedes Wort auf einer Zeile) angezeigt.
- die zweite Datei wird "1" genannt. Es erfüllt alle Spezifikationen der ersten Datei.
- Die restlichen Dateien erfüllen alle Spezifikationen der ersten Datei und repräsentieren den Rest der Ziffern.

Bei der Standardausgabe wird nichts angezeigt. Alle 10 Wörterbuchdateien werden erstellt, auch wenn einige davon leer sind.

## Aufgabe 8

Eine Liste von Dateien wird als Parameter auf der Befehlszeile bereitgestellt, die jeweils ein Transport-Ticket darstellen. Ein solches Ticket kann „IN“ sein, was bedeutet, dass die Waren in das Lager eintreten, oder „OUT“, was bedeutet, dass die Waren das Lager verlassen. Der Aufbau eines Transport-Tickets ist wie folgt: In der ersten Zeile der Datei steht die Art des Transports („IN“ oder „OUT“), in den restlichen Zeilen stehen Informationen zu den transportierten Produkten. Jedes Produkt befindet sich auf einer Linie, gefolgt von seiner Menge. Die Informationen werden durch ein Komma abgetrennt. Die Datei „lager.csv“ enthält den Bestand der Produkte im Lager, strukturiert im gleichen Format. Jedes Produkt steht auf einer Linie, gefolgt von der Menge aus dem Lager. Aktualisieren Sie die Daten in der „lager.csv“-Datei nach der Bearbeitung der Transport-Tickets. Bei der Standardausgabe wird nichts angezeigt.

### Beispiel für „lager.csv“-Datei:

Zement, 4

Montagekleber, 9

### Beispiel für Transportdatei:

IN

Zement, 12

## Aufgabe 9

In einem Programm vom Typ ".c" ist ein Schreibfehler aufgetreten, und der Parameter einer Funktion muss durch einen bestimmten Wert ersetzt werden. Schreiben Sie ein Skript, das für eine bestimmte Funktion bei jedem Aufruf den Funktionsparameter durch einen bestimmten Wert ersetzt. Das Skript empfängt über den ersten Parameter in der Befehlszeile der Ordner, in dem sich das Programm befindet, über den zweiten Parameter den Namen der Funktion und über den dritten Parameter den Wert, durch den es ersetzt werden muss. Das Skript zeigt in der Standardausgabe die Anzahl der vorgenommenen Ersetzungen an.

## Aufgabe 10

Ein Finanzberatungsunternehmen benötigt ein Skript, um einen Berechnungsprozess zu automatisieren. Für jeden Kunden gibt es eine nach dem Kunden benannte Datei mit der Erweiterung ".txt". Die Datei enthält in der ersten Zeile eine Zahl, die das Gehalt des Kunden (monatliches Einkommen) darstellt, und in der zweiten Zeile andere numerische Werte, die die monatlichen Ausgaben dieses Kunden darstellen. Daten werden durch Leerzeichen abgetrennt. Das Unternehmen möchte für jeden Kunden den Notfallfonds und die Anzahl der Monate berechnen, in denen er angesammelt werden kann. Der Notfallfonds stellt den Betrag dar, der die Ausgaben für einen Zeitraum von 6 Monaten abdeckt. Schreiben Sie ein Skript, das eine Liste mit Kundennamen als Parameter in der Befehlszeile enthält. Für jeden Kunden wird am Ende seiner persönlichen Datei eine Zeile hinzugefügt, die die Höhe des Notfallfonds und die Anzahl der Monate, in denen er angesammelt werden kann, enthält. Bei der Standardausgabe wird nichts angezeigt.