

## Shell Übungsaufgaben Blatt 1

**Aufgabe 1:** Erstelle eine Datei deren Erstellungsdatum vor 4 Jahren war.

**Aufgabe 2:** Schreibe ein Shell-Skript, das wenn an einem Sonntag ausgeführt wird “Du sollst am Sonntag nicht arbeiten” ausgibt.

**Aufgabe 3:** Erstelle eine Shell-Funktion, die wenn sie aus der Shell aufgerufen wird, die verbleibenden Stunden bis 18 Uhr ausgibt.

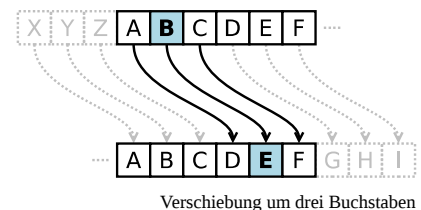
**Aufgabe 4:** Schreibe ein Shell-Skript das sich selber (den Code) ausgibt.

**Aufgabe 5:** Schreibe ein Shell-Skript, das beliebig viele (per Parameter definierbare) zufällige Wörter ausgibt.

**Aufgabe 6:** Sortiere die Wörter aus Aufgabe 5.

**Aufgabe 7:** Wie gut ist dein Zufallsgenerator: Überprüfe ab welcher Anzahl von zufallsgenerierten Wörtern, sich die Wörter wiederholen.

**Aufgabe 8:** Schreibe ein Shell-Skript das die Ausgabe aus Aufgabe 5 Cäsar (Verschiebechiffre) verschlüsselt und wieder entschlüsselt. Lese den Shift über einen Shellparameter ein.



**Aufgabe 9:** Sortiere deine shell history und gib die meist genutzten Befehle aus

## Shell Übungsaufgaben Blatt 2

**Aufgabe 10:** Erstelle ein Shell-Skript das ein 'sicheres' Password generiert

**Aufgabe 11:** Erstelle ein Shell-Skript das fuer alle eingeloggten User, die laufenden Prozesse ausliest und deren offene Files ausgibt.

**Aufgabe 12:** Erstelle ein git-repository auf [git.uibk.ac.at](https://git.uibk.ac.at) und checke deine Skripte dort ein

**Aufgabe 13:** Erstelle ein Shell-Skript, das die ProzessID und alle offenen Filedeskriptoren des Skript inclusive deren Nummer ausgibt.

**Aufgabe 14:** Erstelle ein Shell-Skript das endlos wartet – wenn das Skript mit Strg+C beendet wird soll “Mit Strg+C beendet”, wenn mit kill das Skript beendet wird soll “Mit kill beendet” ausgegeben werden.

**Aufgabe 15:** Schreibe ein Shell-Skript das dir deine lokalen IP Adresse(n) und die externe IP Adresse (mit der Du im Internet surfst) ausgibt.

**Aufgabe 16:** Schreibe ein Shell-Skript das Ports eines Servers bis zum HTTPS Port (443) scannt und ausgibt ob diese offen oder geschlossen sind.

## Shell Aufgaben (einfach zu lösen in der Shell) aus den Beispielen der Python Übungsblättern

### Kontrollfluss (Übungsblatt 1)

1. Geben Sie die Zahlen von 25 bis 1 rückwärts aus (25, 24, . . . , 1).
2. Geben Sie alle Zahlen von 1 bis inklusive 100 aus, die entweder durch 3 oder durch 7 teilbar sind.
3. Geben Sie die ersten  $n$  Fibonacci-Zahlen aus. Es gilt  $F_n = F_{n-1} + F_{n-2}$ ,  $F_1=1$  und  $F_0=0$ . Die Variable  $n$  wird vorher beliebig gesetzt (Beispiel:  $n = 42$ ).
4. Geben Sie alle Vielfachen der Zahlen 2 bis inklusive 10 aus, die kleiner als 100 sind. Beginnen mit den Vielfachen von 2: 2, 4, . . . , 98. Dann kommen alle Vielfachen von 3 (3, 6, . . . , 99), und so weiter.

### Maps (Übungsblatt 2)

1. Implementieren Sie mit Hilfe einer Map ein kleines Telefonbuch. Verwenden Sie den Namen einer Person als Schlüssel, und die Telefonnummer als Wert. Implementieren Sie eine Funktion `printContactInfo(phonebook, name)`, mit der das Telefonbuch durchsucht werden kann. Gibt es einen Eintrag unter dem Namen `name`, werden Name und Telefonnummer auf der Kommandozeile ausgegeben. Gibt es keinen Eintrag unter diesem Namen, geben Sie eine Fehlermeldung aus (etwa: "No such name"). Implementieren Sie auch eine Funktion, mit der alle Kontakte ausgegeben werden können!
2. Welche Probleme könnten bei der vorgeschlagenen Lösung auftreten, und wie würden Sie diese umgehen?

### Passwörter (Übungsblatt 3)

1. Schreiben Sie ein Programm, mit dem zufällige Passwörter generiert werden können. Welche Zeichenklassen (Großbuchstaben, Ziffern, Sonderzeichen, . . . ) zugelassen sind, sollte konfigurierbar sein. Implementieren Sie auch eine Funktion, mit der Sie beliebig viele Passwörter beliebiger Länge generieren können.

### Textanalyse (Übungsblatt 3)

1. Schreiben Sie ein Programm, um Texte zu analysieren. Normalisieren Sie den Text zuerst, indem Sie Sonderzeichen eliminieren und den Text in Kleinbuchstaben umwandeln. Zählen Sie, wie oft jedes Wort im Text vorkommt. Implementieren Sie auch Funktionen, um alle Wörter nach Häufigkeit sortiert auszugeben, und um die  $n$  häufigsten Wörter auszugeben.
2. Testen Sie Ihr Programm mit dem Buch "Pride and Prejudice" von Jane Austen.  
Download: <http://www.gutenberg.org/files/1342/1342-0.txt>