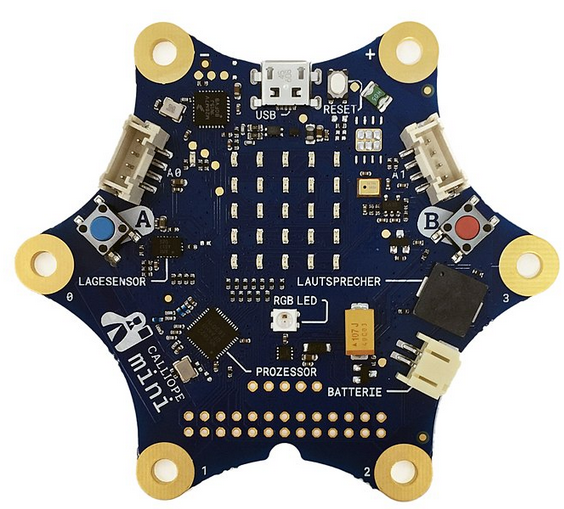
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



Hacking als Beruf

# Stufe 1: Calliope Mini als Tresor-Schloss

Das „Passwort“ ist eine Folge von vier Pins, die in der richtigen Reihenfolge berührt werden müssen. Ein Pin (Nummer 0, 1, 2 oder 3) kann im Passwort mehrmals auftreten, es ist also zum Beispiel auch das Passwort „0 0 2 0“ möglich.

Um das Raten einfacher zu machen, gibt der Calliope nach jedem Pin Feedback, ob der richtige Pin an dieser Stelle gedrückt wurde.

**Aufgabe:**

Findet das Passwort des Calliope durch Probieren heraus und protokolliert die Versuche, bis es geknackt wurde.

*Vorgehen*: Eine Hand berührt den Minus-Pin, die andere Hand berührt einen der Pins 0 bis 3. Bei Erfolg blinkt eine entsprechende LED und man „rät“ den nächsten Pin. Bei Misserfolg beginnt das Raten wieder mit der 1. Pin. Notiert bei Erfolg die Anzahl der Versuche, die der Calliope euch anzeigt. Ein neues Passwort wird mit Taste B generiert. Wechselt euch ab und wiederholt das Ganze zehn Mal.

Wie viele Versuche werden im Mittel gebraucht, um das Passwort zu knacken?

**Denkaufgaben (zu zweit)**

1. Wie viele Versuche werden im schlimmsten Fall gebraucht, um das Passwort zu knacken?
2. Wie viele Versuche werden im schlimmsten Fall gebraucht, um ein fünfstelliges Passwort zu knacken (auch mit 4 mögliche Nummern pro Pin)?
3. Wie viele Versuche werden (im schlimmsten Fall) gebraucht, um ein Passwort mit ***s*** Stellen zu knacken, wobei jede Pin ***m*** verschiedenen Möglichkeiten haben kann?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



Hacking als Beruf

# Stufe 2: Programmieren mit Python

Zur Vorbereitung auf die nächste Stufe (Knacken von Passwörtern in Python) werden hier die wichtigsten Grundlagen der Programmiersprache Python erklärt:

**Aufgabe**

Schreibt ein Python-Programm, welches für jede Zahl zwischen 0 und 99 auf den Bildschirm schreibt, ob die Zahl gerade ist oder nicht.

Ausgabe auf dem Bildschirm: 0 gerade

1. ungerade
2. gerade

…

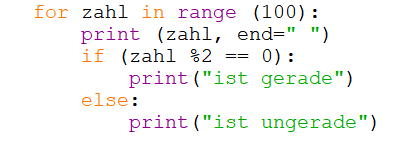
**Dinge wiederholen (Schleife):**

Gehe alle Zahlen durch von 0 bis 99: Beginne im ersten Durchlauf mit **zahl**=0. Die Schleife erhöht **zahl** um 1 in jedem neuen Durchlauf. Schreibe jedes Mal den Wert von **zahl** auf den Bildschirm:



**Bedingungen formulieren:**

Wenn die Zahl gerade ist, dann schreibe „gerade“ hinter die Zahl, sonst schreibe „ungerade“:



Ändere dein Programm, so dass nun ausgegeben wird, ob die Zahl durch 7 teilbar ist oder nicht.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ein Bild, das Vogel, draußen, groß, Bär enthält.

Automatisch generierte Beschreibung

Hacking als Beruf

# Stufe 3: Passwort knacken mit Python

Das „Passwort“ ist eine vierstellige Zahl (alle 10 Ziffern 0...9 sind pro Stelle erlaubt). Eine Ziffer kann im Passwort mehrmals auftreten, es ist also zum Beispiel auch die Passwort-Zahl „1 0 1 0“ möglich.

Achtung: Anders als beim Calliope weiß man hier erst, ob das Passwort stimmt, wenn alle vier Stellen richtig geraten wurden.

**Aufgabe (gemeinsames Programmieren als Black1 und Black2):**

Schreibt ein Programm zum Knacken des Passworts und zählt (im Programm) die Versuche, bis es geknackt wurde.

Lasst ein neues Passwort (aus vier Zufallszahlen) erstellen (in der Shell **reset\_zahl()** aufrufen) und lasst euer Programm erneut laufen. Wiederholt dies zehn Mal. Protokolliert die Anzahl der Versuche bei jedem Lauf. Wie viele Versuche werden im Mittel gebraucht, um das Passwort zu knacken?

**Denkaufgaben (zu zweit)**

1. Wie viele Versuche werden im schlimmsten Fall gebraucht, um das Passwort zu knacken?
2. Wie viele Versuche werden im schlimmsten Fall gebraucht, um ein fünfstelliges Passwort zu knacken (auch mit 10 mögliche Ziffern pro Pin)?
3. **Knackig**: Wie viele Versuche werden (im schlimmsten Fall) gebraucht, um ein Passwort mit ***s*** Stellen zu knacken, wobei jede Pin ***m*** verschiedenen Möglichkeiten haben kann?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ein Bild, das Vogel, draußen, groß, Bär enthält.

Automatisch generierte Beschreibung

Hacking als Beruf

# Stufe 4: Passwort knacken mit Python (Buchstabenkombinationen)

Das „Passwort“ ist eine vierstellige Buchstabenkombination (erlaubt sind kleine Buchstaben). Ein Buchstabe kann im Passwort mehrmals auftreten, es ist also zum Beispiel auch das Passwort abba“ möglich.

Achtung: Anders als beim Calliope weiß man hier erst, ob das Passwort stimmt, wenn alle vier Stellen richtig geraten wurden.

**Aufgabe**

Schreibt ein Programm zum Knacken des Passworts und zählt (im Programm) die Versuche, bis es geknackt wurde. Lasst ein neues Passwort (aus vier kleinen Buchstaben) erstellen (in der Shell **reset\_wort()** aufrufen) und lasst euer Programm erneut laufen. Wiederholt dies zehn Mal. Protokolliert die Anzahl der Versuche bei jedem Lauf. Wie viele Versuche werden im Mittel gebraucht, um das Passwort zu knacken?

**Denkaufgaben (zu zweit)**

1. Wie viele Versuche werden im schlimmsten Fall gebraucht, um das Passwort zu knacken?
2. Wie viele Versuche werden im schlimmsten Fall gebraucht, um ein fünfstelliges Passwort zu knacken (auch mit 26 möglichen Kleinbuchstaben pro Stelle)?
3. Wie viele Versuche werden (im schlimmsten Fall) gebraucht, um ein Passwort mit ***s*** Stellen zu knacken, wobei jede Stelle ***m*** verschiedenen Möglichkeiten haben kann?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



Hacking als Beruf

# Stufe 5:

# Schneller hacken: Wörterbuch und persönliche Informationen einbeziehen

Das „Passwort“ ist wieder eine vierstellige Buchstabenkombination (erlaubt sind kleine Buchstaben). Ein Buchstabe kann im Passwort mehrmals auftreten, es ist also zum Beispiel auch das Passwort „*abba*“ möglich.

**Aufgabe**

Erweitert euer Hacking-Programm aus Stufe 4, so dass es

* zunächst alle Wörter aus einem Wörterbuch (Datei woerterbuch\_10000woerter.txt) testet,
* danach bestimmte Wörter testet, die ihr selbst zusätzlich in die Wörterbuch-Datei schreibt. Diese Wörter stammen z.B. aus Onlineprofilen der Person (facebook, instagram, …),
* am Ende wieder alle möglichen Wörter der Länge 4 generiert (Stufe 4) und testet

Durch dieses mehrstufige Verfahren, das persönliche Vorlieben und menschliche Strategien, sich ein Passwort auszudenken, mit einbezieht, steigt die Wahrscheinlichkeit, das Passwort schneller zu knacken als beim sturen *Brute-Force*-Verfahren.

Wörterbuchdateien gibt es im Internet. Das Wörterbuch-Verfahren ist nicht schwer zu programmieren. Es lässt sich auch einfach erweitern, z.B. auf Kombinationen von zwei Wörtern aus der Wörterbuch-Datei.

Dies ist der Grund, warum ihr keine persönlichen Namen und keine einfachen Wörter oder Wörterkombinationen als Passwort nehmen solltet.