

Aufgabenblatt 3

Aufgabe 1. Erste Übung.

- a) Entwickeln Sie einen Algorithmus, der Folgendes leistet:
 - Es wird eine Zahl eingegeben.
 - Wenn die Zahl nicht 42 ist, wird zu der Zahl gezählt:
 - Bsp. 1, Eingabe 45, Ausgabe: 45 44 43 42
 - Bsp. 2, Eingabe 38, Ausgabe: 38 39 40 41 42
- b) Implementieren Sie den Algorithmus in Python.
- c) Definieren Sie Testfälle und testen Sie Ihr Programm mit diesen Testfällen.

Aufgabe 2. Ein kleines Spiel.

Ein Algorithmus soll Folgendes leisten:

- Der Computer „denkt“ sich eine Zahl zwischen 1 und 10.
 - Nutzer:innen geben Zahlen ein, der Computer gibt aus, ob die Zahl größer oder kleiner ist.
 - Ist die eingegebene Zahl gleich, gibt der Computer die Anzahl der Versuche aus, die benötigt wurden, um die Zahl zu erraten.
- a) Entwickeln Sie den Algorithmus.
 - b) Implementieren Sie den Algorithmus in Python. Recherchieren Sie dazu, wie sie zufällige Integer-Zahlen in Python generieren können.
 - c) Überlegen Sie sich, wie sie den Algorithmus systematisch testen können. Erklären Sie Ihre Teststrategie.

Aufgabe 3. Noch ein Spiel.

Programmieren Sie das Nimm-Spiel in Python. Sie haben eine Zahl an Streichhölzern. Die Spieler:innen ziehen wechselseitig mindestens 1 und maximal 3 Hölzchen. Wer das letzte Hölzchen nehmen muss, hat verloren.

- a) Entwickeln Sie einen Algorithmus, der das Spiel mit zwei menschlichen Spielern umsetzt. Die Anzahl der liegenden Hölzchen wird als Zahl ausgegeben.
- b) Implementieren Sie den Algorithmus in Python.
- c) Testen Sie den Algorithmus.
- d) Ändern Sie jetzt das Programm, der zweite Mitspieler soll der Computer sein.
 - a. In der ersten Version soll der Computer immer zufällig ziehen.
 - b. In der zweiten Version sollen die letzten Züge des Computers so erfolgen, dass er gewinnt (kann):

Anzahl	7	6	5	4	3	2
Computer zieht	3	2	1	2	2	1

- c. In der dritten Version soll er folgende Gewinnstrategie umsetzen:
 - i. Wenn vier Hölzchen liegen, ziehe drei.
 - ii. Ansonsten Sorge dafür, dass die Anzahl der liegenden Hölzchen nach deinem Zug durch vier teilbar ist.
 - iii. Ist dies nicht möglich, ziehe eine zufällige Zahl von Hölzchen.

Aufgabe 4. Dreiecke und Bäume.

Recherchieren Sie, wie Sie in Python ein weiteres Zeichen an eine Zeichenkette anhängen können.

Entwickeln Sie Algorithmen, die folgende Figuren ausgeben (Leerzeichen sind durch „_“ dargestellt). Die Nutzer:innen sollen angeben können, wie hoch die Figuren sind.

Implementieren Sie diese Algorithmen in Python.

```
*
**
***
****
*****
*****
```

```
      *
     **
    ***
   ****
  *****
 *****
```

```
      *
     ***
    *****
   *****
  *****
 *****
 *****
```