Übungsblatt 3

Thomas Graf
EF / WF Informatik 2018-2019
Programmieren in Python I

16. September 2018

1 Jede dritte Zahl ($\star$)

Um den Umgang mit `for`-loops in Python nochmals zu üben, informiere Dich über die `range()`-Funktion auf:
https://www.w3schools.com/python/ref_func_range.asp.
Schreibe (mit Hilfe des neu Gelernten) einen `for`-loop, der jede dritte ganze Zahl, beginnend mit 8 bis 150 ausgibt. Die erste Zahl im Output sollte 8 sein, die letzte 149.

2 Lesen im Skript ($\star\star$)

Lies das Skript bis zum Ende des Abschnitts *Funktionen in Python*. Hast du dazu Fragen? Gibt es Unklarheiten?
Bemerkung:
Listen und jump-statements werden wir im Unterricht noch genauer miteinander anschauen.

3 079... ($\star\star$)

Eine Person verrät uns lediglich die Vorwahl ihrer 10-stelligen Mobiltelefonnummer. Damit gibt es nur noch 10 Millionen Kombinationen, die es auszuprobieren gilt.
Schreibe ein Python-Programm, welches die 2000 kleinsten Nummern auflistet. Die erste Nummer sollte 0790000000 sein, die letzte 0790001999. Das Programm soll die Nummern als Strings ausgeben.

4 Vermutung bestätigen (⋆⋆⋆)

Beim Untersuchen der Zahlenfolge $a_n = n^3 - n$, $n \geq 2$, haben wir zufälligerweise beobachtet, dass für alle von uns ausprobierten Werte von $n \geq 2$, die Zahlen $n^3 - n$, stets durch 3 teilbar zu sein scheinen.
Deshalb wagen wir folgende Behauptung aufzustellen:

$$3 \text{ ist ein Teiler von } n^3 - n \text{ für alle } n \in \mathbb{N},\ n \geq 2. \tag{1}$$

Da wir noch sehr unsicher sind, ob unsere Behauptung 1 tatsächlich stimmt, schreiben wir ein Python-Programm, welches die Behauptung bis zu $n = 1000$ überprüfen soll.

a)
Schreibe solch ein Python-Programm und folgere daraus, dass Behauptung 1 möglicherweise tatsächlich stimmen könnte.

Der Test in Python hat unsere Vermutung bestärkt. Nun möchten wir versuchen, unsere Behauptung zu beweisen.

b)
Beweise Behauptung 1 mittels vollständiger Induktion.

5 Schachbrett (⚛)

Wir möchten ein quadratisches Schachbrettmuster mit $n \times n$ Feldern erzeugen (für **gerades** $n \geq 2$).
Jedes der n^2 Felder soll dabei eine Grösse von genau $s \times s$ Zeichen haben ($s \geq 1$). Die Zahl `1` repräsentiere die schwarzen Felder, die Zahl `0` die weissen Felder.

Zusätzlich möchten wir wählen können, ob das linke obere Feld schwarz oder weiss sein soll (`upper_left = 'black'` oder `upper_left = 'white'`).
Folgende Beispiele zeigen die entsprechenden Schachbrettmuster für verschiedene Wahlen der drei Parameter `n`, `s` und `upper_left`:

```
# Schachbrettmuster für n = 2, s = 3, upper_left = 'white'
000111
000111
000111
111000
```

```
111000
111000

# Schachbrettmuster für n = 4, s = 1, upper_left = 'black'
1010
0101
1010
0101

# Schachbrettmuster für n = 6, s = 2, upper_left = 'black'
110011001100
110011001100
001100110011
001100110011
110011001100
110011001100
001100110011
001100110011
110011001100
110011001100
001100110011
001100110011
```

Allgemein soll das Muster immer genau $n \cdot s$ Zeichen breit und ebenso hoch sein.

Definiere eine Python-Funktion
`def draw_chessboard(n, s, upper_left)`,
welche die drei oben beschriebenen Parameter akzeptiert und das entsprechende Schachbrettmuster im Terminal ausgibt.

Abgabe: bis spätestens Montag, 17. September 2018, um 08:00.