
Einführung in die Programmierung - WS 2016/2017

4. Übungsblatt

Abgabe: 25.11.2016 - 10:00

Aufgabe 1: Schachbrettmuster

(4 Punkte)

Schreiben Sie ein Programm, das mit Hilfe der Ausgabe von einer bestimmten Anzahl an Sternen(*) und Leerzeichen ein Schachbrett-ähnliches Muster der Größe $n \times n$ erzeugt. Dabei ist n ein Wert, der vom Benutzer eingegeben wird. Ein Beispiel für ein 5×5 Schachbrettmuster finden Sie hier:

```
1 * * * * *
2 * * * * *
3 * * * * *
4 * * * * *
5 * * * * *
```

Aufgabe 2: Collatz-Folge

(6 Punkte)

Die Collatz-Folge ist eine Zahlenfolge, die sich wie folgt bilden lässt: Starte mit einer beliebigen natürlichen Zahl $n > 0$. Ist n gerade, dann ist die nächste Zahl der Folge $n/2$. Ist n ungerade, dann ist die nächste Zahl der Folge $3n + 1$. Startet man mit $n = 20$, so erhält man diese Folge 10, 5, 16, 8, 4, 2, 1. Ihre Aufgabe ist nun wie folgt:

- Entwerfen Sie ein Flow-Chart zum Konstruieren einer Collatzfolge. Dabei soll der Benutzer die natürliche Zahl n eingeben können und die Collatzfolge ausgegeben werden. Eine Übersicht über die Elemente eines Flow-Chart sowie ein Beispiel dazu finden Sie am Ende des Übungsblatts.
- Implementieren Sie anschließend das Programm. Das Programm sollte zum Flow-Chart passen!

Übrigens...

Die Collatz-Folge geht auf den deutschen Mathematiker Lothar Collatz zurück, der 1937 auch eine berühmte und bisher nicht bewiesene Vermutung aufstellte, die auch als *Collatz-Problem* bekannt ist: *Unabhängig vom Startwert, mündet jede Collatz-Folge im Zyklus 4, 2, 1.*

Aufgabe 3: Wachstumsverhalten von Funktionen

(4 Punkte)

Schreiben Sie ein Programm, das für die Werte $n = 2, 4, 8, 16, \dots, 2048$ jeweils folgende Funktionswerte in einer Zeile ausgibt:

- $\log_2(n)$
- n
- $n \cdot \log_e(n)$
- n^2
- n^3
- 2^n

Um die einzelnen Werte voneinander zu trennen, können Sie Tabs (" $\backslash t$ ") verwenden.

Aufgabe 4: Ramanujan Taxi

(6 Punkte)

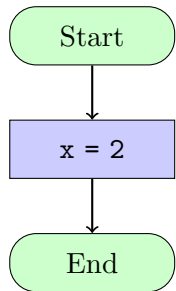
S. Ramanujan war ein indischer Mathematiker, der für sein Gespür für Zahlen berühmt war. So existiert folgende Anekdote über ihn: Einmal war der englische Mathematiker G. H. Hardy auf dem Weg um Ramanujan im Krankenhaus zu besuchen. Das Taxi, in dem er saß um zum Krankenhaus zu gelangen, hatte die Nummer 1729. Bei Ramanujan angekommen, bezeichnete Hardy die Nummer als “rather dull number” was so viel “eher langweilige Zahl” bedeutet. Ramanujan widersprach ihm und sagte, dass diese Nummer eine sehr interessante Zahl sei, da sie die kleinste Zahl ist, die als Summe von zwei kubischen Zahlen auf zwei unterschiedlichen Wegen ausgedrückt werden kann.

Schreiben Sie ein Programm, das zu einer Zahl $n \in \mathbb{N}$ alle Zahlen $z \leq n$ ausgibt, die auf mindestens zwei Arten als Summe von zwei kubischen Zahlen ausgedrückt werden können. D.h. finden Sie alle Zahlen z , für die gilt: $\exists a \neq b \neq c \neq d : z = a^3 + b^3 = c^3 + d^3, a, b, c, d \in \mathbb{N}$

Verwenden Sie dazu vier verschachtelte **for**-Schleifen.

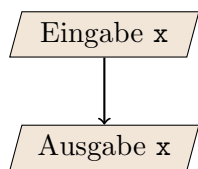
Notationen von Flow-Charts:

Anfang/Ende



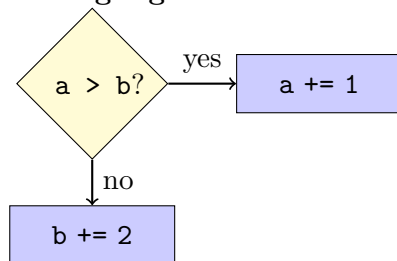
```
1 x = 2
```

Eingabe/Ausgabe



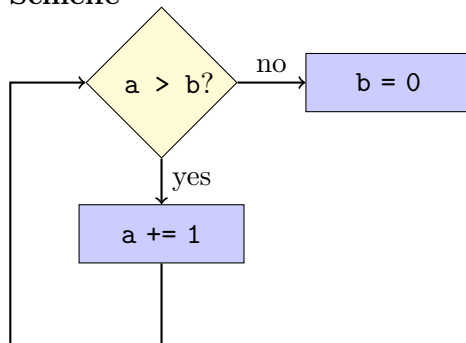
```
1 x = int(input())  
2 print(x)
```

Bedingung



```
1 if a > b:  
2     a += 1  
3 else:  
4     b += 2
```

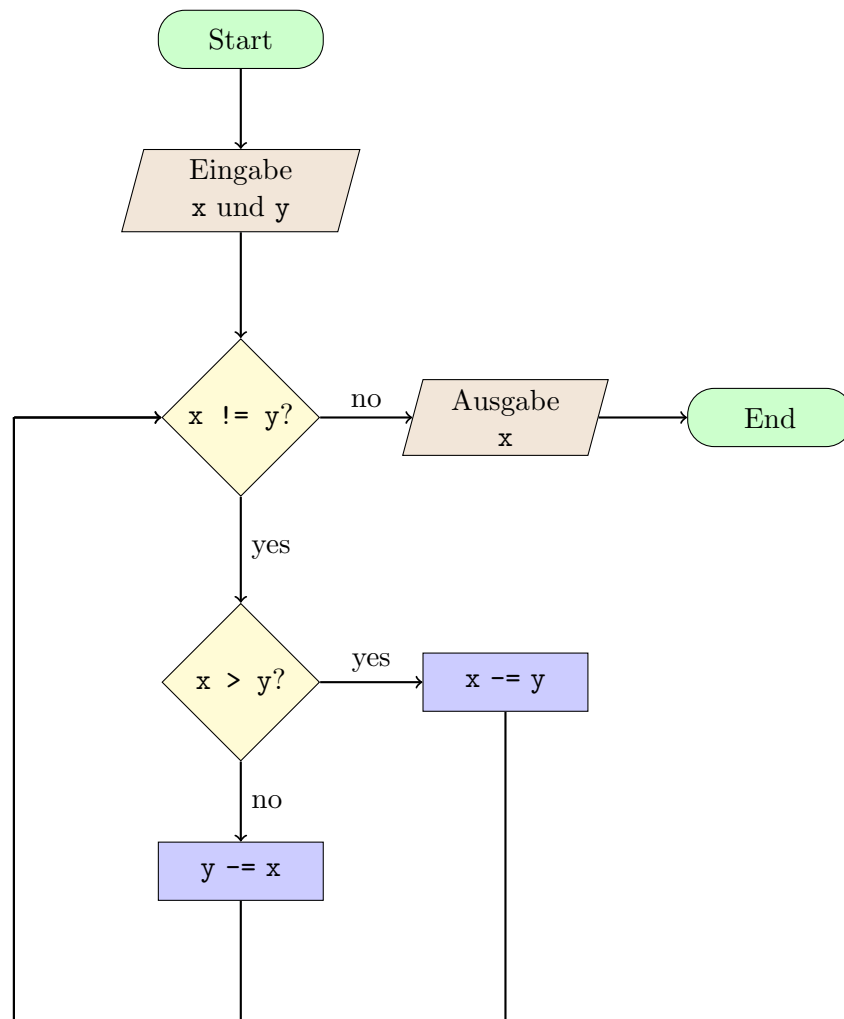
Schleife



```
1 while a > b:  
2     a += 1  
3     b = 0
```

Beispiel zum Erstellen von Flow-Charts:

In der Vorlesung haben Sie den Euklidischen Algorithmus zum Berechnen des größten gemeinsamen Teilers(ggT) kennengelernt. Hier finden Sie nun ein Beispiel, bei dem der Nutzer zwei Zahlen x und y eingibt. Anschließend wird deren ggT berechnet und ausgegeben.



```
1 x = int(input("a ="))
2 y = int(input("b ="))
3
4 while x != y:
5     if x > y:
6         x -= y
7     else:
8         y -= x
9 print(x)
```