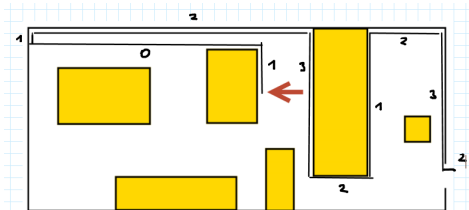
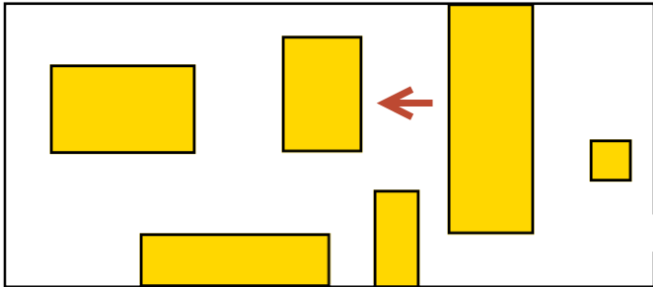


1. (3 Punkte) Notiere das Ergebnis und die Folge der Zahlen, die während des Collatz-Algorithmus berechnet werden für die folgende Eingabe: 14.

Lösung:

7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
Ergebnis: 17

2. (4 Punkte) Zeichne den Weg aus dem Labyrinth nach dem Pledge-Algorithmus und notiere die Abfolge des Zählers. Der Pfeil zeigt Startpunkt und Bewegungsrichtung zu Beginn.



Lösung:

Zähler: 0 1 0 1 2 3 2 1 2 3 2

3. (3 Punkte) Ermittle mit dem klassischen Euklidischen Algorithmus den größten gemeinsamen Teiler von 68 und 12. Notiere den Ablauf des Algorithmus nach dem Verfahren aus dem Unterricht und gib das Ergebnis an.

Lösung:

68	12
56	12
44	12
32	12
20	12
8	12
8	4
4	4

Der ggT ist 4.

4. (3 Punkte) Ermittle mit dem modernen Euklidischen Algorithmus den größten gemeinsamen Teiler von 244 und 16. Notiere den Ablauf des Algorithmus nach dem Verfahren aus dem Unterricht und gib das Ergebnis an.

Lösung:

244	16
16	4
4	0

Der ggT ist 4.

5. (2 Punkte) Beim Sieb des Eratosthenes wurde gerade die 3 endgültig zur Primzahl erklärt und der rechte Finger hat die darauf folgende Streichaktion beendet. Bei der anschließenden Übertragung des Listeninhaltes hat sich ein Fehler eingeschlichen. Welcher Index zeigt auf den falschen Wert? (T = True, F = False).

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
T	T	T	T	F	T	F	T	F	F	F	T	F	T	F	F	F	T	F	T	F	F	F	F	F	T	F	F	F	T	F

Lösung: 23

6. (3 Punkte) Beschreibe in Worten, welchen Wert die Funktion *doit* ermittelt. Was wird im angegebenen Beispiel ausgegeben?

```
def doit(a):
    best = None
    best_val = -float('inf')
    for i in range(len(a)-1):
        val = a[i+1]
        if val > best_val:
            best_val = val
            best = i
    return best
```

```
a = [50,32,66,94,42,14,77,89]
print(doit(a))
```

Lösung:

Die Funktion *doit* führt eine lineare Suche auf der Liste *a* durch. Sie ermittelt den Index der Zahl, die den größten Nachfolger hat. Die Ausgabe ist: 2

7. (4 Punkte) Die Aufgabe bezieht sich auf die Implementation der binaeren Suche aus dem Unterricht. Die binaere Suche besteht aus (in der Regel) mehreren Durchgängen, in denen immer wieder der linke oder rechte Zeigefinger (*i* und *j*) versetzt wird, bis in der Mitte die gesuchte Zahl gefunden wurde oder die Finger sich überkreuzt haben.

Beispiel: Bei Suche nach $x = 16$ in der Liste $a = [4, 7, 12, 13, 19]$ stehen die Variablen *i*, *j*, *mitte* zu Beginn auf 0, 4, 2. Am Ende des ersten Durchgangs haben die Variablen die Werte auf 3, 4, 3.

```
x = 42
a = [4, 7, 10, 11, 13, 17, 21, 24, 27, 29, 30, 32, 40, 45]
binaereSuche(a,x)
```

Zu Beginn eines Durchgangs sollen die Variablen *i*, *j*, *mitte* die Werte 7 13 10 haben.

- Welche Werte haben sie am Ende des Durchgangs?
- Wird ein weiterer Durchgang folgen?

Lösung: a. 11 13 12 b. Ja