

Aufgabe 1 (2P)

- Undo-Funktion in Anwendungsprogrammen (1 P)
- Prüfen, ob ein Klammerterm korrekt ist (1 P)
- History-Funktion in Browsern
- ...

Aufgabe 2 (2P)

```
class Stack:
```

```
    def __init__(self):
        self.items = []

    def push(self, item):
        self.items.append(item) # 0.5P

    def pop(self):
        return self.items.pop() # 0.5P

    def peek(self):
        return self.items[-1]   # 0.5P

    def is_empty(self):
        return self.items == [] # 0.5P
```

Aufgabe 3 (2P)

```
from stack import Stack
```

```
s = Stack()
s.push(3)
s.push(5)
s.push(1)
a = s.pop()
b = s.peek()
s.push(8)
```

```
print(a)
print(b)
print(s.size())
```

```
1
5
3
```

Aufgabe 4 (2P)

(a) ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~4~~ 5 ~~6~~ ~~7~~ 8 ~~9~~
 2 1 3 0 4 6 7 9 False

(b) ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~9~~
 2 3 1 4 6 5 8 7 0 9 True

Aufgabe 5 (2P)

(a) M D - C F + * ⇒ (M - D) * (C + F)

(b) - * P S + X Y ⇒ P * S - (X + Y)

Aufgabe 6 (4P)

Input:

P	-	(	C	+	F	)	*	A	-	E	+	W	*	M
--------------	--------------	--------------	--------------	--------------	--------------	--------------	--------------	--------------	--------------	--------------	--------------	--------------	--------------	--------------

Stack: (nimmt → zu)

-	(	+	)	*	-	+	*							
--------------	--------------	--------------	--------------	--------------	--------------	--------------	--------------	--	--	--	--	--	--	--

Output:

P	C	F	+	A	*	-	E	-	W	M	*	+		
---	---	---	---	---	---	---	---	---	---	---	---	---	--	--

Aufgabe 7 (2P)

Input:

3	5	2	+	6	4	+	*	2	-	+
---	---	---	---	---	---	---	---	---	---	---

Stack: (nimmt $\rightarrow$ zu)

3	5	2	7	6	4	10	70	2	68	71
---	---	---	---	---	---	----	----	---	----	----

Output: **71**

Aufgabe 8 (2P)

(a) Ein Algorithmus ist ein Lösungsverfahren, das

- endlich
- deterministisch (der nächsten Schritt ist eindeutig bestimmt)
- effektiv (die Wirkung jeder Anweisung ist eindeutig)

ist.

- (b)
- Handelt es sich um ein Lösungsverfahren? Ja
 - Ist es endlich? Ja
 - Ist es deterministisch? Nein, da es keine exakten Vorgabe gibt, wie die Artikel zu suchen sind.
 - Ist es effektiv? Nein, da Artikel möglicherweise im Angebot des Supermarktes fehlen und nicht vorgegeben ist, was in diesem Fall zu tun ist.

Aufgabe 9 (2P)

$$\begin{aligned} \text{ggT}(30, 12) &= (18, 12) = (6, 12) \stackrel{s}{=} (12, 6) = (6, 6) \\ &= (0, 6) \stackrel{s}{=} (6, 0) = (6, 0) = 6 \end{aligned}$$

Aufgabe 10 (2P)

$$\text{ggT}(32, 18) = (18, 14) = (14, 4) = (4, 2) = (2, 0) = 2$$

Aufgabe 11 (2P)

```
def function(n):  
    s = 0                                # 1*c  
    for i in range(0, n):                # n*c  
        for j in range(0, n):            # n*n*c  
            s += i + j                    # 3*n*n*c  
        for k in range(0, 2*n):          # 2*n*n*c  
            s += k                        # 4*n*n*c  
    return s                             # 1*c
```

Summe: $T(n) = 10cn^2 + cn + 2c \in O(n^2)$

Aufgabe 12 (1P)

$$T(n) \in O(n^2) \quad \Rightarrow \quad T(n) = C \cdot n^2$$

$$T(100) = C \cdot 100^2 = 3 \text{ s}$$

$$T(300) = C \cdot 300^2 = 9 \cdot C \cdot 100^2 = 9 \cdot 3 \text{ s} = 27 \text{ s}$$