

1. (3 Punkte) Die rekursive Funktion `sternedoppelt` gibt einen String zurück, der jedes Zeichen des übergebenen Strings durch `'**'` ersetzt.

Beispiel:

```
>>> sternedoppelt('lol')
'*****'
```

Lösung:

```
def sternedoppelt(s):
    if len(s) == 0:
        return ''
    return '**' + sternedoppelt(s[1:])
```

2. (3 Punkte) Die rekursive Funktion `tausche` vertauscht je zwei Zeichen eines Strings. Falls die Anzahl der Zeichen ungerade ist, bleibt das letzte Zeichen an seinem Platz. Implementiere diese Funktion.

```
>>> tausche('abcde')
'badce'
>>> tausche('a')
'a'
```

Lösung:

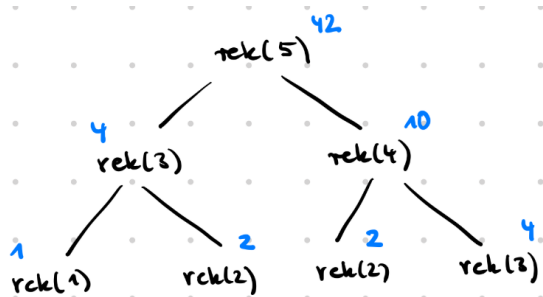
```
def tausche(s):
    if len(s) <= 1: return s
    return s[1]+s[0]+tausche(s[2:])
```

3. (3 Punkte) Gib einen geeigneten Rekursionsbaum zur Ermittlung der Zwischenergebnisse an. Was erscheint auf der Konsole?

```
def rek(n):
    if n <= 2: return n
    return 2+rek(n-2)*rek(n-1)

print(rek(5))
```

Lösung:



Die Ausgabe ist 42.

4. (3 Punkte) Gib die Abfolge der rekursiven Aufrufe an. Was erscheint auf der Konsole?

```
def rek(a,b):
    if b == 0: return a+1
    return rek(2*b,a%b)

print(rek(15,20))
```

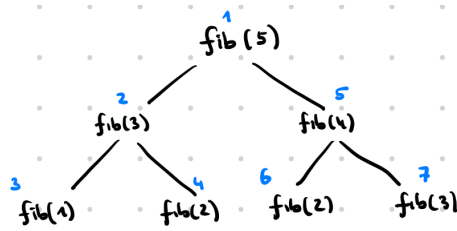
Lösung:

```
rek(15,20)
rek(40,15)
rek(30,10)
rek(20,0)
Die Ausgabe ist 21.
```

5. (2 Punkte) Die rekursive Funktion fib wird mit fib(5) aufgerufen. Wie heißt der 6. und 7. Aufruf von fib?

```
def fib(n):
    if n <= 2: return 1
    return fib(n-2) + fib(n-1)
```

Lösung:

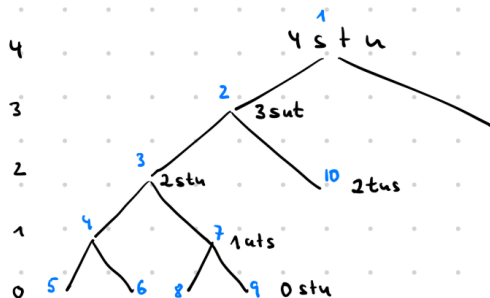


```
6. fib(2)
7. fib(3)
```

6. (2 Punkte) Die rekursive Funktion hanoi wird mit hanoi(4,'s','t','u') aufgerufen. Wie heißen der 9. und 10. Aufruf von hanoi?

```
def hanoi(n, start, ziel, zwischen):
    if n == 0: return
    hanoi(n-1, start, zwischen, ziel)
    print("Scheibe", n, "von", start, "nach", ziel)
    hanoi(n-1, zwischen, ziel, start)
```

Lösung:



```
9. hanoi(0, 's', 't', 'u')
10. hanoi(2, 't', 'u', 's')
```