

1. (2 Punkte) Schreibe eine Funktion `func01`, der ein String übergeben wird und die den String ohne erstes und letztes Zeichen zurückgibt.

```
>>> func01('Hallo')  
'all'
```

2. (3 Punkte) Schreibe eine Funktion `func02`, der ein nicht leerer String übergeben wird und die den String wieder zurückgibt, aber erstes und letztes Zeichen sind vertauscht.

```
>>> func02('Hallo')  
'oallH'
```

3. (3 Punkte) Schreibe eine Funktion `func03`, der ein String übergeben wird und die den String wieder zurückgibt ergänzt um den selben String in umgekehrter Reihenfolge.

```
>>> func03('Hallo')  
'HalloollaH'
```

4. (3 Punkte) Schreibe eine Funktion `zaehl01`, die einen nicht leeren String erhält und zurückgibt, wieviel mal das erste Zeichen in dem String vorkommt. Die Stringmethode `count` darf nicht benutzt werden.

```
>>> zaehl01('abbaeacde')  
4
```

5. (3 Punkte) Schreibe eine Funktion `zaehl02`, die einen nicht leeren String erhält und zurückgibt, wieviele Ziffern (0-9) in dem String vorkommen.

```
>>> zaehl02('ab12cc3d60bb')
5
```

6. (3 Punkte) Schreibe eine Funktion `zaehl03`, die einen nicht leeren String und ein Zeichen erhält und zurückgibt, wie oft das Zeichen an einem geraden Index des Strings vorkommt.

```
>>> zaehl03('abaabbbba', 'a')
3
>>> zaehl03('abaabbbba', 'b')
2
```

7. (4 Punkte) Schreibe eine Funktion `zaehl01`, die zurückgibt, wieviele Zahlen es zwischen 100 und 999 (einschließlich) gibt, die durch 3 aber nicht durch 5 teilbar sind.

8. (4 Punkte) Schreibe eine Funktion `zaehl02`, der zwei natürliche Zahlen übergeben werden und die zurückgibt, wieviele Vielfache von 7 zwischen diesen Zahlen (einschließlich) liegen.

```
>>> zaehl02(4,17)
2
>>> zaehl02(0,21)
4
```

9. (2 Punkte) Schreibe eine Funktion `pruef01`, der ein String mit Länge  $\geq 2$  übergeben wird und die prüft, ob die ersten beiden Zeichen gleich sind.

```
>>> pruef01('aab')
True
>>> pruef01('abb')
False
```

10. (4 Punkte) Schreibe eine Funktion `pruef02`, der ein String `s`, ein Zeichen `c` und eine nicht negative ganze Zahl `k` übergeben wird, und die prüft, ob `c` in `s` `k`-mal vorkommt.

```
>>> pruef02('aab', 'a', 2)
True
>>> pruef02('aab', 'b', 2)
False
```

11. (3 Punkte) Schreibe eine Funktion `summe01`, die die Summe aller Vielfachen von 7 zwischen 100 und 1000 zurückgibt.

12. (4 Punkte) Schreibe eine Funktion `echteTeilerSumme`, der eine positive ganze Zahl übergeben wird und die die Summe der echten Teiler dieser Zahl zurückgibt. Echte Teiler sind alle Teiler der Zahl, außer die Zahl selbst. Beispiel: die Summe der echten Teiler von 12 ist:  $1 + 2 + 3 + 4 + 6 = 16$

```
>>> echteTeilerSumme(12)
16
```

13. (4 Punkte) Was erscheint auf der Konsole?

```
def a(x):
    return x + 1
def b(x):
    return x + 2
def c(x,y):
    return x + y
def d(x,y):
    return x > y
z = c(a(3),b(5))
print(d(z,10),z)
```

14. (4 Punkte) Was erscheint auf der Konsole?

```
def a(x,y):
    return x + 2*y
def b(x):
    return x - 1
def c(x,y):
    return x - y
def d(x,y):
    return x - y > 0
z = c(a(3,4),b(2))
print(d(z,12),z)
```