Python Lösungen+ Programmierpraxis (Kurzversion)

Aufgabe 1

```
resultat = ((93.46-86.39)/(97.42+48.64)-8.93)**2

resultat = round(resultat, 2)

print(resultat) # => 78.88
```

Aufgabe 2

```
import math

resultat = math.sqrt(60.37+6.41*80.46/(69.47+40.04))

resultat = round(resultat, 2)

print(resultat) # => 8.07
```

Aufgabe 3

```
import math

resultat = math.exp(-0.5*((18.65-19.74)/17.15))

resultat = round(resultat, 2)

print(resultat) # => 1.0
```

Aufgabe 4

```
import math

resultat = -3/4*math.log(1-4/3*0.64)

resultat = round(resultat, 2)

print(resultat) # => 1.44
```

Aufgabe 5

```
b = bin(6409)

print(b) # => 0b1100100001001
```

Aufgabe 6

```
d = int('1101110111', 2)

print(d) # => 887
```

Aufgabe 7

```
h = hex(2020)

print(h) # => 0x7E4
```

Aufgabe 8

```
d = int('D02C0FA7', 16)

print(d) # => 3492548519
```

Aufgabe 9 (leicht)

```
A = [4, 7, 8, 1, 9, 3]
B = []

n = len(A)
for i in range(0, n):
    B.append(A[n-i-1])

print(A)
print(B)
```

Aufgabe 10 (leicht)

```
L = []
while True:
    eingabe = input('Geben sie eine Zahl ein: ')
    if eingabe == '':
        break
    else:
        zahl = float(eingabe)
        L.append(zahl)

print(L)
```

Aufgabe 11 (leicht)

```
def gendern(wort, typ='s'):
    '''Ergänzt 'wort um ein Gener-Suffix. 's'=singular, 'p'=plural'''
    if typ == 's':
        return wort + '*in'
    elif typ == 'p':
        return wort + '*innen'
    else:
```

```
        print(f'Gib "s" (singular) oder "p" (plural) statt {typ} ein')
        return wort + '?'

if __name__ == '__main__': # Testcode

    print(gendern('Radfahrer'))
    print(gendern('Schüler', 's'))
    print(gendern('Pilot', 'p'))
    print(gendern('Akademiker', 'x'))
```

Aufgabe 12 (leicht)

```
infile = 'werte_in_yard.txt'
outfile = 'werte_in_metern.txt'

def yard_to_meter(x):
    return x*0.9144

dd_in  = open(infile,  mode='r')
dd_out = open(outfile, mode='w')

for line in dd_in:
    try:
        meter = yard_to_meter(float(line))
        dd_out.write('{0:.3f}\n'.format(meter))
    except ValueError:
        print('Skipping line ...')

dd_in.close()
dd_out.close()
```

Aufgabe 13 (leicht)

```
def summe_datei(dateiname):
    dd = open(dateiname)
    summe = 0
    for zeile in dd:
        summe += float(zeile)
    dd.close()
    return summe

def summe_datei_plus(dateiname):
    dd = open(dateiname)
    summe = 0
    for zeile in dd:
        try:
            summe += float(zeile)
        except ValueError:
            print('Überspringe Zeile ...')

    dd.close()
```

```
    return summe

s1 = summe_datei('numbers.txt')
s2 = summe_datei_plus('numbers_dirty.txt')
print('summe_datei(...):      ', s1)
print('summe_datei_plus(...):', s2)
```

Aufgabe 14 (leicht)

```
L = [4, 7, 1, 2, 8, 2]

summe = 0
for x in L:
    summe += x

if len(L) == 0:
    print('Der Mittelwert der leeren Liste ist nicht definiert.')
else:
    mw = summe/len(L)
    print(mw)
```

Aufgabe 15 (leicht)

```
L = [3,7,2,1,8,4,6]

m = L[0]
for i in range(1, len(L)):
    if L[i] > m:
        m = L[i]

print(m, L)
```

Aufgabe 16 (mittel)

```
def ziffernhaeufigkeit(number):
    freq = [0] * 10

    for c in str(number):
        if c in '0123456789':
            freq[int(c)] += 1

    n = sum(freq)
    for i in range(0, 10):
        freq[i] /= n

    return freq

print(ziffernhaeufigkeit(12121212))
```

Aufgabe 17 (mittel)

```python
def schaltjahr(jahr):
    '''Gibt True zurück, wenn 'jahr' ein Schaltjahr ist und False sonst.'''
    if jahr % 400 == 0:
        return True
    elif jahr % 100 == 0:
        return False
    elif jahr % 4 == 0:
        return True
    else:
        return False
```

Aufgabe 18 (mittel)

```python
N = ['A', 'C', 'C', 'A', 'T', 'G', 'T']
K = []

for x in N:
    if x == 'A':
        K.append('T')
    elif x == 'C':
        K.append('G')
    elif x == 'G':
        K.append('C')
    elif x == 'T':
        K.append('A')
    else:
        K.append('?')
        print(f'{x} ist kein gültiges Nukleotid.')

print('N:', N)
print('K:', K)
```

Aufgabe 19 (mittel)

```python
def median(liste):
    '''Gibt den Median von 'liste' zurück.'''
    L = sorted(liste) # erzeuge eine neue Liste
    n = len(L)
    if n % 2 == 1:
        return L[n//2]
    else:
        return (L[n//2-1] + L[n//2])/2

if __name__ == '__main__': # Testcode

    L = [8, 7, 4, 9] # => [4, 7, 8, 9] => 7.5
    print(median(L))

    L = [5, 1, 5, 6, 3] # => [1, 3, 5, 5, 6]  => 5
    print(median(L))
```

Aufgabe 20 (mittel/schwierig)

```
def kennzahlen(dateiname):
    data = []
    dd = open(dateiname, mode='r')
    for zeile in dd:
        try:
            data.append(float(zeile))
        except ValueError:
            print('Überspringe Zeile ...')

    x_min = min(data)
    x_max = max(data)
    R = x_max- x_min
    n = len(data)
    m = sum(data)/n
    v = sum([(x - m)**2 for x in data])/(n-1)

    print('Anzahl Werte: {0}'.format(n))
    print('Minimum: {0}'.format(x_min))
    print('Maximum: {0}'.format(x_max))
    print('Spannweite: {0}'.format(R))
    print('empirischer Mittelwert: {0}'.format(m))
    print('empirische Varianz: {0}'.format(v))

kennzahlen('numbers.txt')
```

Aufgabe 21 (mittel/schwierig)

```
def eratosthenes(n):
    sieb = [1 for i in range(0, n)]
    sieb[0], sieb[1] = 0, 0
    i = 2
    while i < int(n**0.5)+1:
        if i != 0:
            for j in range(i*i, n, i):
                sieb[j] = 0
        i += 1
    return sieb


fd = open('primzahlen.txt', mode='w')
sieb = eratosthenes(36)
for i in range(0, len(sieb)):
    if sieb[i] == 1:
        fd.write(f'{i}\n')
fd.close()
```

Aufgabe 22 (schwierig)

```
def wechselgeld(preis, zahlung):
    '''Gibt die Anzahl der Münzen eines Wechselgelds zurück.'''
```

```
    muenzen = [500, 200, 100, 50, 20, 10, 5]
    anzahl = []
    n = len(muenzen)
    differenz = zahlung - preis
    if differenz < 0:
        return "Noch {0} Rappen nachzahlen!\n".format(-differenz)

    for i in range(0, n):
        anzahl.append(differenz // muenzen[i])
        differenz %= muenzen[i]

    txt  = 'Zahlung: {0} Rappen\n'.format(zahlung)
    txt += 'Preis:   {0} Rappen\n'.format(preis)
    txt += 'Retour:  {0} Rappen\n'.format(zahlung-preis)
    for i in range(0,n):
        if anzahl[i] > 0:
            txt += '{0} x {1} Rappen\n'.format(anzahl[i], muenzen[i])

    return txt

if __name__ == '__main__':

    print(wechselgeld(55, 500))
    print(wechselgeld(420, 410))
    print(wechselgeld(375, 375))
    print(wechselgeld(210, 500))
```