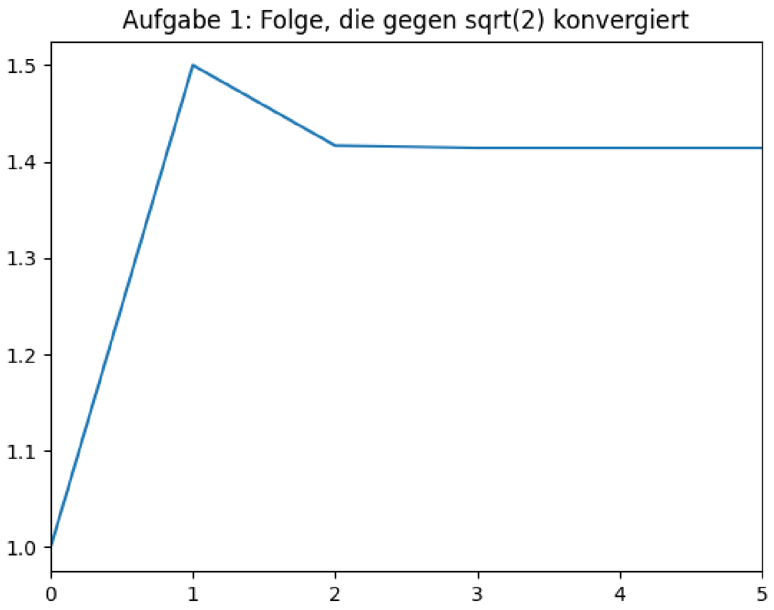
```python
from math import *
import matplotlib.pyplot as plt

def task1(a) -> None:
    print("--------Aufgabe 1--------")
    x_n = [1]
    a *= a  # !ACHTUNG: Ungenauigkeit von Gleitkommazahlen!
    print(f"a: {a}")

    for i in range(20):
        x_n1 = 0.5 * (x_n[i] + a / x_n[i])
        x_n.append(x_n1)
        print("i: {i:2} = {x_n:.30f}".format(i=i, x_n=x_n[i]))

    draw_task1(x_n)


def draw_task1(y) -> None:
    x = [i for i in range(len(y))]
    plt.xlim(0, 5)
    plt.plot(x, y)
    plt.title("Aufgabe 1: Folge, die gegen sqrt(2) konvergiert")
    plt.show()
```



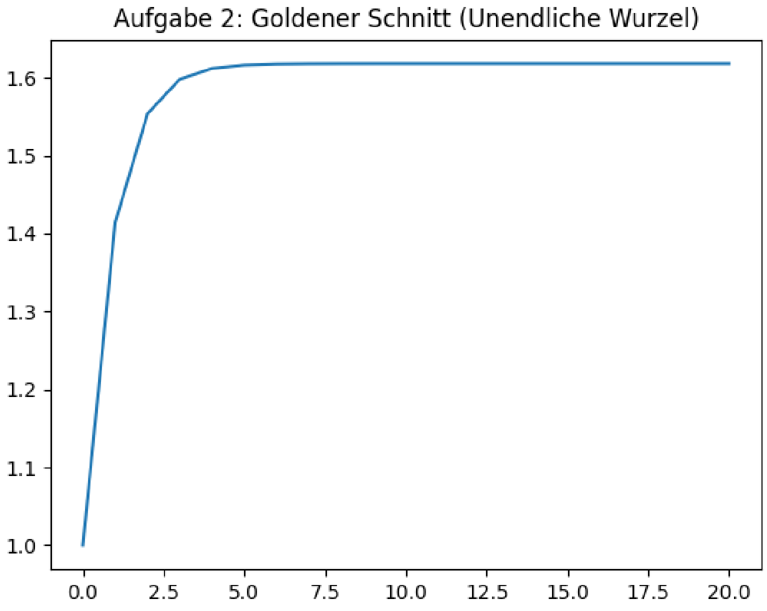
```
--------Aufgabe 1--------
a: 2.0000000000000004
i:  0 = 1.000000000000000000000000000000
i:  1 = 1.500000000000000222044604925031
i:  2 = 1.416666666666666962726139900042
i:  3 = 1.414215686274510108688673426514
i:  4 = 1.414213562374690091871798358625
i:  5 = 1.414213562373095367519226783770
i:  6 = 1.414213562373095367519226783770
i:  7 = 1.414213562373095367519226783770
i:  8 = 1.414213562373095367519226783770
i:  9 = 1.414213562373095367519226783770
i: 10 = 1.414213562373095367519226783770
i: 11 = 1.414213562373095367519226783770
i: 12 = 1.414213562373095367519226783770
i: 13 = 1.414213562373095367519226783770
i: 14 = 1.414213562373095367519226783770
i: 15 = 1.414213562373095367519226783770
i: 16 = 1.414213562373095367519226783770
i: 17 = 1.414213562373095367519226783770
i: 18 = 1.414213562373095367519226783770
i: 19 = 1.414213562373095367519226783770
```

```
def task2() -> None:
    print("\n--------Aufgabe 2--------")
    a_n = [1]

    for n in range(20):
        a_n1 = sqrt(1 + a_n[n])
        a_n.append(a_n1)
        print("n: {n:2} = {a_n:.30f}".format(n=n, a_n=a_n[n]))

    draw_task2(a_n)


def draw_task2(y) -> None:
    x = [i for i in range(len(y))]
    plt.plot(x, y)
    plt.title("Aufgabe 2: Goldener Schnitt (Unendliche Wurzel)")
    plt.show()
```

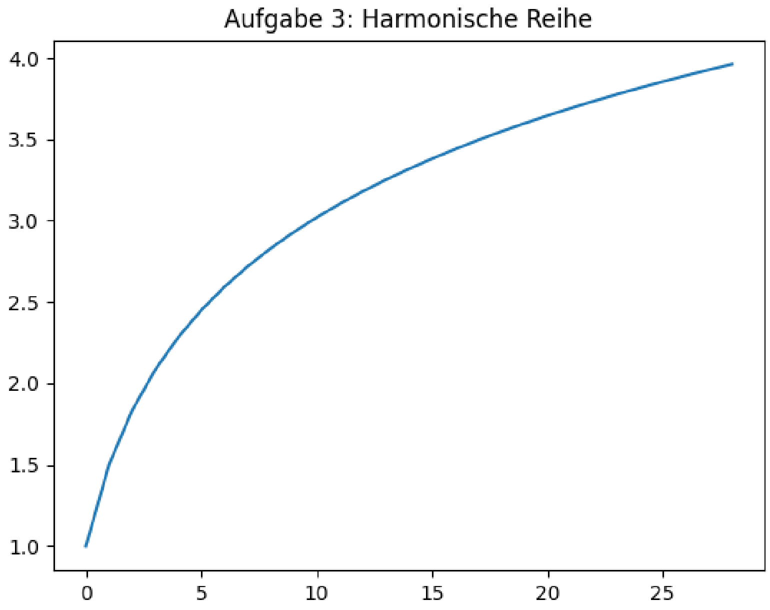


```
--------Aufgabe 2--------
n:  0 = 1.000000000000000000000000000000
n:  1 = 1.414213562373095145474621858739
n:  2 = 1.553773974030037363647238635167
n:  3 = 1.598053182478617495476669319032
n:  4 = 1.611847754125251608314783879905
n:  5 = 1.616121206508116969757793413009
n:  6 = 1.617442798527390479534915357362
n:  7 = 1.617851290609674919096505618654
n:  8 = 1.617977530934739283097201223427
n:  9 = 1.618016542231487608916040699114
n: 10 = 1.618028597470232421429159330728
n: 11 = 1.618032322751999929622002127871
n: 12 = 1.618033473928150778320400604571
n: 13 = 1.618033829661218891970975164440
n: 14 = 1.618033939588789715813277325651
n: 15 = 1.618033973558277827109463942179
n: 16 = 1.618033984055426977022307255538
n: 17 = 1.618033987299224429889932252991
n: 18 = 1.618033988301613046445481813862
n: 19 = 1.618033988611368156895764514047
```

```
def task3() -> None:
    print("\n--------Aufgabe 3--------")
    a_n = [1]
    i = 0
    for n in range(2, 30):
        print("n: {n:2} = {a_n:.15f}".format(n=n, a_n=a_n[i]))
        a_n1 = a_n[i] + 1 / n
        a_n.append(a_n1)
        i += 1

    draw_task3(a_n)


def draw_task3(y) -> None:
    x = [i for i in range(len(y))]
    plt.plot(x, y)
    plt.title("Aufgabe 3: Harmonische Reihe")
    plt.show()
```

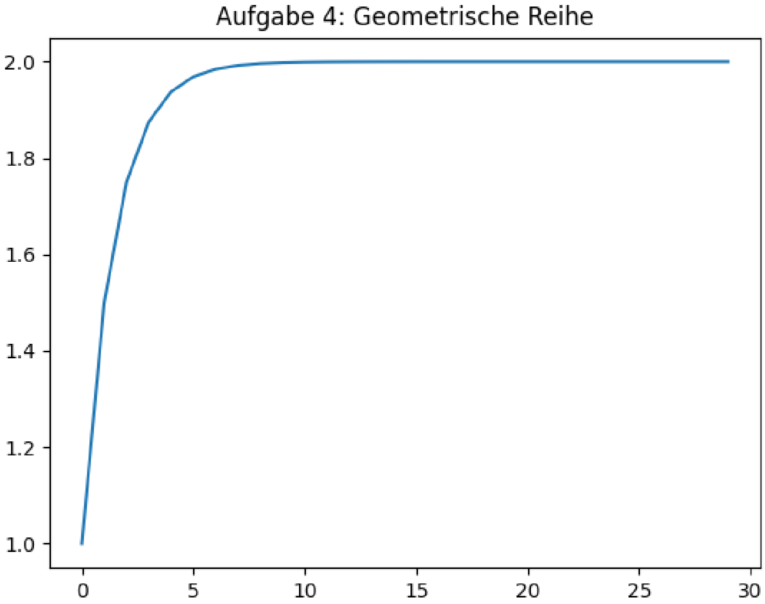


```
--------Aufgabe 3--------
n:  2 = 1.000000000000000
n:  3 = 1.500000000000000
n:  4 = 1.833333333333333
n:  5 = 2.083333333333333
n:  6 = 2.283333333333333
n:  7 = 2.450000000000000
n:  8 = 2.592857142857143
n:  9 = 2.717857142857143
n: 10 = 2.828968253968254
n: 11 = 2.928968253968254
n: 12 = 3.019877344877345
n: 13 = 3.103210678210678
n: 14 = 3.180133755133755
n: 15 = 3.251562326562327
n: 16 = 3.318228993228994
n: 17 = 3.380728993228994
n: 18 = 3.439552522640758
n: 19 = 3.495108078196314
n: 20 = 3.547739657143682
n: 21 = 3.597739657143682
n: 22 = 3.645358704762729
n: 23 = 3.690813250217275
n: 24 = 3.734291511086840
n: 25 = 3.775958177753507
n: 26 = 3.815958177753507
n: 27 = 3.854419716215045
n: 28 = 3.891456753252082
n: 29 = 3.927171038966368
```

```python
def task4() -> None:
    print("\n--------Aufgabe 4--------")
    a_n = [1]
    i = 0
    for n in range(1, 30):
        print("n: {n:2} = {a_n:.15f}".format(n=n, a_n=a_n[i]))
        a_n1 = 1 / 2 ** n
        a_n1 += a_n[i]
        a_n.append(a_n1)
        i += 1

    draw_task4(a_n)


def draw_task4(y) -> None:
    x = [i for i in range(len(y))]
    plt.plot(x, y)
    plt.title("Aufgabe 4: Geometrische Reihe")
    plt.show()
```

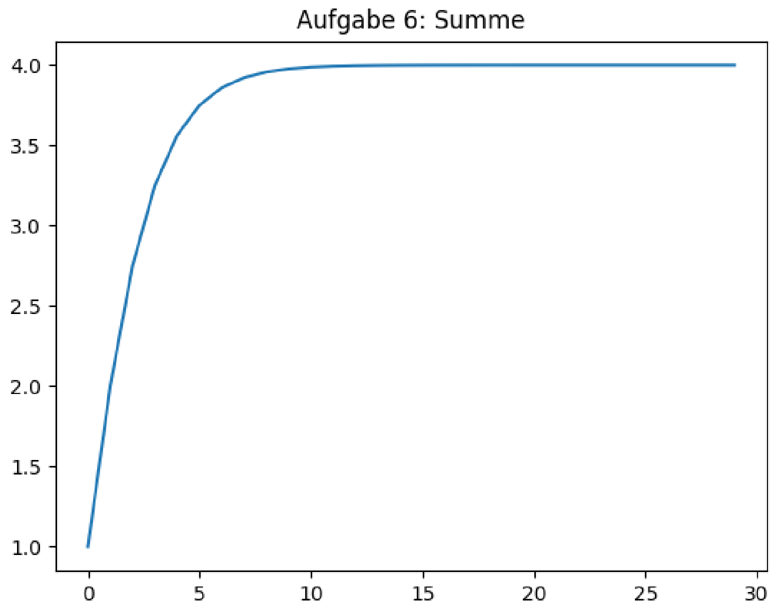


```
--------Aufgabe 4--------
n:  1 = 1.000000000000000
n:  2 = 1.500000000000000
n:  3 = 1.750000000000000
n:  4 = 1.875000000000000
n:  5 = 1.937500000000000
n:  6 = 1.968750000000000
n:  7 = 1.984375000000000
n:  8 = 1.992187500000000
n:  9 = 1.996093750000000
n: 10 = 1.998046875000000
n: 11 = 1.999023437500000
n: 12 = 1.999511718750000
n: 13 = 1.999755859375000
n: 14 = 1.999877929687500
n: 15 = 1.999938964843750
n: 16 = 1.999969482421875
n: 17 = 1.999984741210938
n: 18 = 1.999992370605469
n: 19 = 1.999996185302734
n: 20 = 1.999998092651367
n: 21 = 1.999999046325684
n: 22 = 1.999999523162842
n: 23 = 1.999999761581421
n: 24 = 1.999999880790710
n: 25 = 1.999999940395355
n: 26 = 1.999999970197678
n: 27 = 1.999999985098839
n: 28 = 1.999999992549419
n: 29 = 1.999999996274710
```

```python
def task6() -> None:
    print("\n--------Aufgabe 6--------")
    a_n = [1]
    i = 0
    for n in range(1, 30):
        print("n: {n:2} = {a_n:.15f}".format(n=n, a_n=a_n[i]))
        a_n1 = (n + 1) * pow(2, -n)
        a_n1 += a_n[i]
        a_n.append(a_n1)
        i += 1

    draw_task6(a_n)

def draw_task6(y) -> None:
    x = [i for i in range(len(y))]
    plt.plot(x, y)
    plt.title("Aufgabe 6: Summe")
    plt.show()
```



```python
task1(sqrt(2))
task2()
task3()
task4()
task6()
```

```
--------Aufgabe 6--------
n:  1 = 1.000000000000000
n:  2 = 2.000000000000000
n:  3 = 2.750000000000000
n:  4 = 3.250000000000000
n:  5 = 3.562500000000000
n:  6 = 3.750000000000000
n:  7 = 3.859375000000000
n:  8 = 3.921875000000000
n:  9 = 3.957031250000000
n: 10 = 3.976562500000000
n: 11 = 3.987304687500000
n: 12 = 3.993164062500000
n: 13 = 3.996337890625000
n: 14 = 3.998046875000000
n: 15 = 3.998962402343750
n: 16 = 3.999450683593750
n: 17 = 3.999710083007812
n: 18 = 3.999847412109375
n: 19 = 3.999919891357422
n: 20 = 3.999958038330078
n: 21 = 3.999978065490723
n: 22 = 3.999988555908203
n: 23 = 3.999994039535522
n: 24 = 3.999996900558472
n: 25 = 3.999998390674591
n: 26 = 3.999999165534973
n: 27 = 3.999999567866325
n: 28 = 3.999999776482582
n: 29 = 3.999999884516001
```