

"""

Created on Sat Jul 15 22:44:57 2023

@author: User2

"""

```
#Best: Reference value
#anz: List size
#diff: List of sub values
#unt: difference resistorlist and wanted resistor
#posunt: positive version of unt (in case it's neg)
#l: run count, not needed cause of min()
#z: position of min value in diff
#zw: value of min difference
#zwpr: percentage of min difference
#db: Lowest available resistor Value
```

Aufgabe 1

```
from math import *
#Liste mit Widerstandswerten
werte = [10, 47, 150, 220, 330, 470, 1000, 1500, 2200, 4700, 10000, 22000]

#Hauptprogramm
soll = int(input(" Sollwert in Ohm: "))
best = werte [0]
anz = len(werte)

diff = []
l = 0
for i in range(0, anz):
    unt = werte [i] - soll
    posunt = abs(unt)
    diff.append(posunt)

#index für die niedrigste Value finden
z = diff.index(min(diff))
zw = min(diff)
zwpr = zw * 100 / soll
db = werte[z]
print("Bester Einzelwiderstand:", db)
print(f"Abweichung: {zw} Ohm ({zwpr:.3})%")
```

Aufgabe 2

```

from math import *
#Liste mit Widerstandswerten
werte = [10, 47, 150, 220, 330, 470, 1000, 1500, 2200, 4700, 10000, 2200

#Hauptprogramm
soll = int (input("Sollwert in Ohm: "))
best = werte [0]
anz = len(werte)
diff = []
l = 0

def widerstand_einzel (soll):
    for i in range(0, anz):
        unt = werte [i] - soll
        posunt = abs(unt)
        diff.append(posunt)

widerstand_einzel(soll)
#kein return hinter der Funktion, weil es soll nichts dannach zurückgege
# sondern eifach mit den negsten rechnungen darunter weitergerechnet wer

#index für die niedrigste Value finden
z = diff.index(min(diff))
zw = min(diff)
zwpr = zw * 100 / soll
db = werte[z]
print("Bester Einzelwiderstand:", db)
print(f"Abweichung: {zw} Ohm ({zwpr:.3})%")

```

%% Aufgabe 3

```

from math import *
#Liste mit Widerstandswerten
werte = [10, 47, 100, 150, 220, 330, 470, 1000, 1500, 2200, 4700, 10000,

#Hauptprogramm
soll = int (input("Sollwert in Ohm: "))

best = werte [0]
anz = len(werte)
diff = []
l = 0

# Funktion widerstand einzel
def widerstand_einzel (soll):
    for i in range(0, anz):

```

```

        unt = werte [i] - soll
        posunt = abs(unt)
        diff.append(posunt)

widerstand_einzel(soll)

#index für die niedrigste Value finden
z = diff.index(min(diff))
zw = min(diff)
zwpr = zw * 100 / soll
db = werte[z]
print("Bester Einzelwiderstand:", db)
print(f"Abweichung: {zw} Ohm ({zwpr:.3})%")


anz = len(werte)

# Alle kombinationen von zwei Widerstandswerten bilden
def widerstand_reihe(soll):
    for r1 in range(0,anz):
        for r2 in range(0, anz):
            unt = werte[r1] + werte[r2] - soll
            posnut = abs(unt)

            diff = max(werte)
            r1no = 0
            r2no = 0

            if posnut < diff:
                diff = posnut
                r1no = r1
                r2no = r2

            r1value = werte[r1no]
            r2value = werte[r2no]
            r_total = r1value + r2value

            print(f"Beste Reihenschaltung: {r1value} und {r2value} Ohm")
            print(f"Gesamtwiderstand: {r_total} Ohm")
            print(f"Abweichung in Ohm: {diff}")
            print(f"Abweichung vom Sollwert %: {diff / soll * 100}%")

widerstand_reihe(soll)

### Aufgabe 4

```

```

from math import *
#Liste mit Widerstandswerten
werte = [10, 47, 100, 150, 220, 330, 470, 1000, 1500, 2200, 4700, 10000,

#Hauptprogramm
soll = int (input("Sollwert in Ohm: "))

best = werte [0]
anz = len(werte)
diff = []
l = 0

# Funktion widerstand einzel
def widerstand_einzel (soll):
    for i in range(0, anz):
        unt = werte [i] - soll
        posunt = abs(unt)
        diff.append(posunt)

#index für die niedrigste Value finden
    z = diff.index(min(diff))
    zw = min(diff)
    zwpr = zw * 100 / soll
    db = werte[z]
    print("Bester Einzelwiderstand:", db)
    print(f"Abweichung: {zw} Ohm ({zwpr:.3})%")
    print(" ")
    return
widerstand_einzel(soll)

# parallell widerstand

anz = len(werte)
lowest_diff = 0
p1no = 0
p2no = 0
def widerstand_paralell(soll):
    lowest_diff = (max(werte)*max(werte))/(max(werte) + max(werte))
    # setzt die differenz am anfang hoch, wird nach jedem durchlauf der
    for r1 in range(0, anz): #ich nehme mir einen Widerstand aus der Box
        #print("Widerstand ", r1, "kombiniert mit:")
        for r2 in range(0, anz): #ich kombiniere meinen Widerstand mit a
            #print("Widerstand", r2)
            rges=(werte[r1]*werte[r2])/(werte[r1]+werte[r2]) #ich rechne
            rdifff=rges-soll
            posrdifff = abs(rdifff)

```

```

        if posrdifff < lowest_diff:
            lowest_diff = posrdifff
            p1no = r1
            p2no = r2
            srges=srges

r1value = werte[p1no]
r2value = werte[p2no]

print(f"Beste Parallelschaltung: {r1value} and {r2value} Ohm.")
print(f"Gesamtwiderstand: {srges:.2f} Ohm")
print(f"Difference: {lowest_diff:.2f} Ohm ({lowest_diff / soll * 100}")
print(" ")
return

widerstand_paralell(soll)

# Alle kombinationen von zwei Widerstandswerten bilden
anz = len(werte)
r1no = 0
r2no = 0
lowest_diff2 = 0
def widerstand_reihe(soll):
    lowest_diff2 = max(werte) + max(werte) - soll
    for r1 in range(0,anz):
        for r2 in range(0, anz):
            unt = werte[r1] + werte[r2] - soll
            posnut = abs(unt)

            if posnut < lowest_diff2:
                lowest_diff2 = posnut
                r1no = r1
                r2no = r2

    r1value = werte[r1no]
    r2value = werte[r2no]
    r_total = r1value + r2value

    print(f"Beste Reihenschaltung: {r1value} und {r2value} Ohm")
    print(f"Gesamtwiderstand: {r_total} Ohm")
    print(f"Abweichung in Ohm: {lowest_diff2}")
    print(f"Abweichung vom Sollwert %: {lowest_diff2 / soll * 100}%")
    return

widerstand_reihe(soll)

```

Zusatzaufgabe 2

```
from math import *
#Liste mit Widerstandswerten
werte = [10, 47, 100, 150, 220, 330, 470, 1000, 1500, 2200, 4700, 10000,

try:
#Hauptprogramm
    soll = int (input("Sollwert in Ohm: "))

    best = werte [0]
    anz = len(werte)
    diff = []
    l = 0

# Funktion widerstand einzel
    def widerstand_einzel (soll):
        for i in range(0, anz):
            unt = werte [i] - soll
            posunt = abs(unt)
            diff.append(posunt)

#index für die niedrigste Value finden
        z = diff.index(min(diff))
        zw = min(diff)
        zwpr = zw * 100 / soll
        db = werte[z]
        print("Bester Einzelwiderstand:", db)
        print(f"Abweichung: {zw} Ohm ({zwpr:.3})%")
        print(" ")
        return
    widerstand_einzel(soll)

# parallell widerstand

    anz = len(werte)
    lowest_diff = 0
    p1no = 0
    p2no = 0
    def widerstand_parallell(soll):
        lowest_diff = (max(werte)*max(werte))/(max(werte) + max(werte))
        # setzt die differenz am anfang hoch, wird nach jedem durchlauf
        for r1 in range(0, anz): #ich nehme mir einen Widerstand aus der
            #print("Widerstand ", r1, "kombiniert mit:")
            for r2 in range(0, anz): #ich kombiniere meinen Widerstand m
```

```

#print("Widerstand", r2)
rges=(werte[r1]*werte[r2])/(werte[r1]+werte[r2]) #ich re
rdiffp=rges-soll
posrdiffp = abs(rdiffp)

if posrdiffp < lowest_diff:
    lowest_diff = posrdiffp
    p1no = r1
    p2no = r2
    srges=rges

```

```

r1value = werte[p1no]
r2value = werte[p2no]

```

```

print(f"Beste Parallelschaltung: {r1value} and {r2value} Ohm.")
print(f"Gesamtwiderstand: {srges:.2f} Ohm")
print(f"Difference: {lowest_diff:.2f} Ohm ({lowest_diff / soll *
print(" ")
return

```

```
widerstand_paralell(soll)
```

```
# Alle kombinationen von zwei Widerstandswerten bilden
```

```

anz = len(werte)
r1no = 0
r2no = 0
lowest_diff2 = 0
def widerstand_reihe(soll):
    lowest_diff2 = max(werte) + max(werte) - soll
    for r1 in range(0,anz):
        for r2 in range(0, anz):
            unt = werte[r1] + werte[r2] - soll
            posnut = abs(unt)

            if posnut < lowest_diff2:
                lowest_diff2 = posnut
                r1no = r1
                r2no = r2

```

```

r1value = werte[r1no]
r2value = werte[r2no]
r_total = r1value + r2value

```

```

print(f"Beste Reihenschaltung: {r1value} und {r2value} Ohm")
print(f"Gesamtwiderstand: {r_total} Ohm")

```

```

        print(f"Abweichung in Ohm: {lowest_diff2}")
        print(f"Abweichung vom Sollwert %: {lowest_diff2 / soll * 100}%")
        return

widerstand_reihe(soll)

except ValueError:
    print("Bitte korrekte Zahl eingeben!")

### Zusatzaufgabe 3

from math import *
#Liste mit Widerstandswerten
werte = [10, 47, 100, 150, 220, 330, 470, 1000, 1500, 2200, 4700, 10000,

try:
    #Hauptprogramm
    soll = int (input("Sollwert in Ohm: "))

    best = werte [0]
    anz = len(werte)
    diff = []
    l = 0
    zw = 0
    # Funktion widerstand einzel
    def widerstand_einzel (soll):
        for i in range(0, anz):
            unt = werte [i] - soll
            posunt = abs(unt)
            diff.append(posunt)

    #index für die niedrigste Value finden
    z = diff.index(min(diff))
    zw = min(diff)
    zwpr = zw * 100 / soll
    db = werte[z]
    print("Bester Einzelwiderstand:", db)
    print(f"Abweichung: {zw} Ohm ({zwpr:.3})%")
    print(" ")
    return
    widerstand_einzel(soll)

# parallell widerstand

```



```

anz = len(werte)
lowest_diff = 0
p1no = 0
p2no = 0
def widerstand_paralell(soll):
    lowest_diff = (max(werte)*max(werte))/(max(werte) + max(werte))
    # setzt die differenz am anfang hoch, wird nach jedem durchlauf
    for r1 in range(0, anz): #ich nehme mir einen Widerstand aus der
        #print("Widerstand ", r1, "kombiniert mit:")
        for r2 in range(0, anz): #ich kombiniere meinen Widerstand m
            #print("Widerstand", r2)
            rges=(werte[r1]*werte[r2])/(werte[r1]+werte[r2]) #ich re
            rdiffp=rges-soll
            posrdiffp = abs(rdiffp)

            if posrdiffp < lowest_diff:
                lowest_diff = posrdiffp
                p1no = r1
                p2no = r2
                srges=rges

    r1value = werte[p1no]
    r2value = werte[p2no]

    print(f"Beste Parallelschaltung: {r1value} and {r2value} Ohm.")
    print(f"Gesamtwiderstand: {srges:.2f} Ohm")
    print(f"Difference: {lowest_diff:.2f} Ohm ({lowest_diff / soll *
    print(" ")
    return

```

```

widerstand_paralell(soll)

```

```

# Alle kombinationen von zwei Widerstandswerten bilden
anz = len(werte)
r1no = 0
r2no = 0
lowest_diff2 = 0
def widerstand_reihe(soll):
    lowest_diff2 = max(werte) + max(werte) - soll
    for r1 in range(0,anz):
        for r2 in range(0, anz):
            unt = werte[r1] + werte[r2] - soll
            posnut = abs(unt)

            if posnut < lowest_diff2:
                lowest_diff2 = posnut

```

```

        r1no = r1
        r2no = r2

    r1value = werte[r1no]
    r2value = werte[r2no]
    r_total = r1value + r2value

    print(f"Beste Reihenschaltung: {r1value} und {r2value} Ohm")
    print(f"Gesamtwiderstand: {r_total} Ohm")
    print(f"Abweichung in Ohm: {lowest_diff2}")
    print(f"Abweichung vom Sollwert %: {lowest_diff2 / soll * 100}%")
    return

widerstand_reihe(soll)

if lowest_diff2 < lowest_diff and lowest_diff2 < zw:
    bestdiff = "reihe"
elif lowest_diff < lowest_diff2 and lowest_diff < zw:
    bestdiff = "paralell"
else:
    bestdiff = "normal"

print(f"Die beste Annäherung an den sollwert bringt: {min(lowest_dif

except ValueError:
    print("Bitte korrekte Zahl eingeben!")

```