

RUNNING TESTS FOR ASSIGNMENT SHEET07

STUDENT IS Freitag

ATTEMPTED TASKS ARE Umdrehen.py
Rechnen.py

STARTING TASK Umdrehen.py

THE ATTEMPTED SOLUTION IS

```
import sys

def umdrehen(input):
    output = []
    for i in range(0, len(input)):
        b = input[-(i+1)]
        for j in range(0, len(b)):
            output += b[-(j+1)]
    return output

quelldateiname = sys.argv[1]

try:
    with open(quelldateiname, 'r') as quelldatei:
        datei = quelldatei.readlines()

    for element in umdrehen(datei):
        print(element, end='')

except :
```

```
    print("Datei", quelldateiname, "konnte nicht gefunden  
werden.")  
except PermissionError:  
    print("Es fehlen die notwendigen Rechte zum Lesen oder  
Schreiben.")
```

THE MASTER SOLUTION IS

```
import sys  
  
text = []  
source = sys.argv[1]  
try:  
    with open(source,"r") as quelle:  
        text = quelle.readlines()  
        for i in range(len(text)-1,-1,-1):  
            for c in range(len(text[i])-1,-1,-1):  
                sys.stdout.write(text[i][c])  
            print()  
except:  
    print("Fehler beim Laden der Datei")
```

THE INPUT DATA CONFIG FOR THIS FILE IS:

```
{  
    "params": {  
        "param1": "file-path"  
    }  
}
```

TEST RUN START HERE

chosen input

"Umdrehen.py" *Das ist ein relativer Dateipfad, der auf eine Datei mit diesem Namen in dem aktuellen Arbeitsverzeichnis zeigt. Diese muss also in jedem Fall und unabhaengig von individuellem Betriebssystem, Ordnerstruktur etc. existieren, nicht-leer und lesbar sein (da es die selbe Datei ist die wir gerade ausfuehren."

your output

```
celinemueller@MacBook-Air-Buro sheet07 % python3 Umdrehen.py Umdrehen.py

)".nebierhcS redo neseL muz ethceR negidnewton eid nelhef sE"(tnirp
:rorrEnoissimreP tpecxe
)".nedrew nednufeg thcin etnnok" ,emanietadlleuq ,"ietaD"(tnirp
:rorrEdnuoFtoNeliF tpecxe

        )'=dne ,tnemele(tnirp
:)ietad(neherdmu ni tnemele rof

)(senildaer.ietadlleuq = ietad
:ietadlleuq sa )'r' ,emanietadlleuq(nepo htiw
:yrt

]1[vgra.sys = emanietadlleuq

tuptuo nruter
])1+j(-[b =+ tuptuo
:))b(nel ,0(egnar ni j rof
])1+i(-[tupni = b
:))tupni(nel ,0(egnar ni i rof
][ = tuptuo
:)tupni(neherdmu fed

sys tropmi%
```

expected output

```

celinemueller@MacBook-Air-Buro sheet07 % python3 Umdrehen.py /Users/celinemueller/Desktop/codeTestEnvironment/Handins/Freitag/sheet07/Umdrehen.py

).nebiehrcS redo neseL muz ethceR negidnewton eid nelhef sE"(tnirp
:rorrEnoissimreP tpecxe
)".nedrew nednufeg thcin etnnok" ,emanietadlleuq ,"ietaD"(tnirp
:rorrEdnuoFtoNeliF tpecxe

        )' '=dne ,tnemele(tnirp
:)ietad(neherdmu ni tnemele rof

)(senildaer.ietadlleuq = ietad
:ietadlleuq sa )'r' ,emanietadlleuq(nepo htiw
:yrt

]l[vgra.sys = emanietadlleuq

tuptuo nruter
])1+j(-[b += tuptuo
:))b(nel ,0(egnar ni j rof
])1+i(-[tupni = b
:))tupni(nel ,0(egnar ni i rof
]l[ = tuptuo
:)tupni(neherdmu fed

sys tropmi
celinemueller@MacBook-Air-Buro sheet07 %

```

note that I gave the complete path to YOUR solution, to enable the comparing of the outputs.

COMMENTS ON YOUR SOLUTION:

Wie du sehen kannst, ist deine Ausgabe mit der der Masterlösung deckungsgleich. Ich kann vielleicht noch anmerken, dass du sehr schön das Dateieinlesen in einen try-except-Block gepackt hast (denn dabei kann viel schief gehen). Du hast schon einen `FileNotFoundError` und einen `PermissionError` (mit schönen Fehlermeldungen) abgefangen. Wenn du möchtest, könntest du am schluss noch einen allgemeinen `except Exception as e` für nicht vorher bekannte Fehler anfügen. Weiterhin ist es sehr schön dass du das Umdrehen der eingelesenen Zeilen in eine Funktion gepackt hast.

STARTING TASK Rechnen.py

THE ATTEMPTED SOLUTION IS

```
import sys

def mean(input):
    temp = 0
    for i in range(len(input)):
        temp += input[i]
    resu = temp/len(input)
    return resu

def min(input):
    liste = input[:]
    l = len(liste)
    for i in range(l):
        min = i
        for j in range(i+1,l):
            if liste[j] < liste[min]:
                min = j
        t = liste[i]
        liste[i] = liste[min]
        liste[min] = t
    return liste[0]

def max(input):
    liste = input[:]
    l = len(liste)
    for i in range(l):
        min = i
        for j in range(i+1,l):
            if liste[j] < liste[min]:
                min = j
        t = liste[i]
        liste[i] = liste[min]
        liste[min] = t
    return liste[-1]

numb = []
```

```

while True:
    try:
        temp = sys.stdin.readline()
        if temp == '':
            break
        else:
            numb.append(float(temp))
    except ValueError:
        print('Bitte nur Zahlen Eingeben')

print(mean(numb))
print(min(numb))
print(max(numb))

```

THE MASTER SOLUTION IS

```

import sys

def avg(zahlen):
    return (sum(zahlen)/len(zahlen))

def min(zahlen):
    min = zahlen[0]
    for i in zahlen:
        if i < min: min = i
    return min

def max(zahlen):
    max = zahlen[0]
    for i in zahlen:
        if i > max: max = i
    return max

z = []
for zeile in sys.stdin:
    z += [float(zeile)]

```

```
print("Durchschnitt:", avg(z))
print("Minimum:", min(z))
print("Maximum:", max(z))
```

THE INPUT DATA CONFIG FOR THIS FILE IS:

```
{
  "stream": [
    {
      "type": "float",
      "repeatable": true,
      "min_repeats": 1,
      "max_repeats": 10
    }
  ]
}
```

TEST RUN START HERE

chosen input

```
[
1,
2.3,
5.5,
11
]
```

your output

```
celinemueller@MacBook-Air-Buro sheet07 % python3 Rechnen.py
1
2.3
5.5
11
4.95
1.0
11.0
celinemueller@MacBook-Air-Buro sheet07 %
```

expected output

```
celinemueller@MacBook-Air-Buro sheet07 % python3 Rechnen.py
1
2.3
5.5
11
Durchschnitt: 4.95
Minimum: 1.0
Maximum: 11.0
```

!Wichtig! Beachte dass, wenn ich die Eingabe über die Standardeingabe manuell mache, ich ein geeignetes Symbol senden muss, um zu signalisieren, dass die Eingabe vorbei ist (das heißt EOF und ist meistens ctrl-d, recherchiere ansonsten für dein System), ansonsten gibt es einen Fehler:

```
celinemueller@MacBook-Air-Buro sheet07 % python3 Rechnen.py
1
2.3
5.5
11
Traceback (most recent call last):
  File "/Users/celinemueller/Desktop/codeTestEnvironment/Mastersolutions/sheet07/Rechnen.py", line 20
, in <module>
    z += [float(zeile)]
           ~~~~~^~~~~~
ValueError: could not convert string to float: '\n'
celinemueller@MacBook-Air-Buro sheet07 %
```

COMMENTS ON YOUR SOLUTION:

Wie du siehst, deckt sich deine Ausgabe mit der der Musterlösung. Du hast die Lösung in Funktionen umgesetzt, und die richtigen Float-

Rechenoperatoren genutzt.

Allerdings muss ich anmerken, dass du für die Berechnung des Minimums und Maximums Funktionen mit einer quadratischen Laufzeit geschrieben hast! Die hier angedeuteten Funktionen eignen sich, um Listen zu sortieren (eine komplexe Aufgabe). Um aber "nur" das Minimum oder Maximum zu finden, reicht es einmal über das Array zu laufen. Bitte überprüfe nochmal, weshalb du hier eine doppelte Schleife notwendig gefunden hast, und wie du es einfacher lösen kannst.
