

# RUNNING TESTS FOR ASSIGNMENT SHEET07

STUDENT IS Ramazanova

ATTEMPTED TASKS ARE Umdrehen.py  
Rechnen.py

STARTING TASK Umdrehen.py

THE ATTEMPTED SOLUTION IS

```
import sys
dateiname=sys.argv[1]
try:
    with open (dateiname, "r") as quelle:
        zeilen=quelle.read()
        print(zeilen[::-1])
except:
    print("Fehler beim Laden der Datei")
```

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

) "ietaD red nedaL mieb relheF" (tnirp
:tpecxe
)] 1-:: [neliez (tnirp
)(daer.elleuq=neliez
:elleuq sa )"r" , emanietad( nepo htiw
:yrt
] 1[vgra.sys=emanietad
sys tropmi
celinemueller@MacBook-Air-Buro sheet07 %
```

## expected output

```
celinemueller@MacBook-Air-Buro sheet07 % python3 Umdrehen.py /Users/celinemueller/Desktop/codeTestEnv/ironment/Handins/Ramazanova/sheet07/Umdrehen.py

) "ietaD red nedaL mieb relheF" (tnirp
:tpecxe
)] 1-:: [neliez (tnirp
)(daer.elleuq=neliez
:elleuq sa )"r" , emanietad( nepo htiw
:yrt
] 1[vgra.sys=emanietad
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. Durch die geschickte Anwendung des Slicing Operators in Kombination mit `.read()` anstatt `.readlines()` hast du es umgangen, in einer doppelten Schleife über die Liste von Strings zu iterieren. Stelle aber sicher dass du es auch manuell könntest.

## STARTING TASK Rechnen.py

## THE ATTEMPTED SOLUTION IS

```

import sys

def durchschnitt(a):
    return(sum(a)/len(a))

def minimum(a):
    minimum=a[0]
    for i in a:
        if i < minimum:
            minimum=i
    return minimum

def maximum(a):
    maximum=a[0]
    for i in a:
        if i > maximum:
            maximum=i
    return maximum

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

print("Durchschnitt:",durchschnitt(z))
print("Minimum:",minimum(z))
print("Maximum:",maximum(z))

```

## 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  
Durchschnitt: 4.95  
Minimum: 1.0  
Maximum: 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. Ich finde es schön, wie du die Funktionen umgesetzt hast, und du hast die richtigen Float-Rechenoperatoren genutzt.

---