

# RUNNING TESTS FOR ASSIGNMENT SHEET07

STUDENT IS Demirel

ATTEMPTED TASKS ARE Umdrehen.py  
Rechnen.py

STARTING TASK Umdrehen.py

THE ATTEMPTED SOLUTION IS

```
import sys
dateiname = sys.argv[1]

a = []
with open(dateiname, "r") as q:
    a = q.readlines()

for line in a:
    line = line.strip('\n')
    print(line[::-1])
```

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
sys tropmi
]l[vgra.sys = emanietad

]l[ = a
:q sa )"r" ,emanietad(nepo htiw
)(senildaer.q = a

:a ni enil rof
)'n\'(pirts.enil = enil
)]l-::[enil(tnirp
celinemueller@MacBook-Air-Buro sheet07 %
```

## expected output

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

)]l-::[enil(tnirp
)'n\'(pirts.enil = enil
:a ni enil rof

)(senildaer.q = a
:q sa )"r" ,emanietad(nepo htiw
]l[ = a

]l[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 man aus den Ausgaben erkennen kann, hat dein Programm korrekt den Inhalt jeder Zeile umgedreht, allerdings die Zeilenreihenfolge an sich bewahrt. Für die Schleife über alle eingelesenen Zeilen bietet sich also eine Schleife an, die Rückwärts über die Zeilen laufen kann. Anstatt `for line in a:` empfehle ich dir also `for i in range(len(a)-1,-1,-1): line = a[i] ...`

## STARTING TASK Rechnen.py

## THE ATTEMPTED SOLUTION IS

```
l = []
a = (float(input("bitte Gib so viele Kommazahlen wie du
willst ein, allerdings beendet 0 das Programm:")))
while a != 0:
    l.append(a)
    a = (float(input("bitte Gib so viele Kommazahlen wie du
willst ein, allerdings beendet 0 das Programm:")))

def avg(l):
    sum = 0
    lenght = len(l)
    for e in l:
        sum += e
    return (sum/lenght)

def mini(l):
    min_n = l[0]
    for e in l:
        if e < min_n:
            min_n = e
    return min_n

def maxi(l):
    max_n = l[0]
    for e in l:
        if e > max_n:
            max_n = e
    return max_n

print(avg(l))
print(mini(l))
print(maxi(l))
```

## 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
bitte Gib so viele Kommazahlen wie du willst ein, allerdings beendet 0 das Programm:1
bitte Gib so viele Kommazahlen wie du willst ein, allerdings beendet 0 das Programm:2.3
bitte Gib so viele Kommazahlen wie du willst ein, allerdings beendet 0 das Programm:5.5
bitte Gib so viele Kommazahlen wie du willst ein, allerdings beendet 0 das Programm:11
bitte Gib so viele Kommazahlen wie du willst ein, allerdings beendet 0 das Programm:0
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. Ich finde es schön, wie du die Funktionen umgesetzt hast, und du hast die richtigen Float-Rechenoperatoren genutzt. Außerdem hast du eine Nutzer-Information am Anfang eingebaut, und das EOF-Problem mit einem Abbruchbefehl (0) umgangen.

---