

RUNNING TESTS FOR ASSIGNMENT SHEET07

STUDENT IS Haag

ATTEMPTED TASKS ARE Umdrehen.py
Rechnen.py

STARTING TASK Umdrehen.py

THE ATTEMPTED SOLUTION IS

```
f=open("Quelldatei.txt", "r")
lines=f.readlines()
f.close()
print(lines)

for i in range(len(lines)-1,-1,-1):
    #print(lines[i])
    line=lines[i]
    for j in range(len(line)-1,-1,-1):
        print(line[j], end="")

#i=0
#while i< len(lines):
#    print(lines[-1-i], end="")
#    i+=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
Traceback (most recent call last):
  File "/Users/celinemueller/Desktop/codeTestEnvironment/Handins/Haag/sheet07/Umdrehen.py", line 1, in <module>
    f=open("Quelldatei.txt", "r")
FileNotFoundError: [Errno 2] No such file or directory: 'Quelldatei.txt'
celinemueller@MacBook-Air-Buro sheet07 %
```

expected output

```
celinemueller@MacBook-Air-Buro sheet07 % python3 Umdrehen.py /Users/celinemueller/Desktop/codeTestEnvironment/Handins/Haag/sheet07/Umdrehen.py
1=+i      #
)""=dne ,li-1-[senil(tnirp      #
:)senil(nel <i elihw#
0=i#

)""=dne ,lj[enil(tnirp
:)1-,1-,1-)enil(nel(egnar ni j rof
]i[senil=enil
)li[senil(tnirp#
:)1-,1-,1-)senil(nel(egnar ni i rof

)senil(tnirp
)(esolc.f
)(senildaer.f=senil
)"r" ,"txt.ietadlleuQ"(nepo=f
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, hast du dein Namen **einer** Quelldatei in das Programm "hard-gecodet", anstatt, wie in der Aufgabenstellung gefordert, den Namen **irgendeiner, vorher nicht definierter** Quelldatei per **Programmparameter** übergeben.

Wenn du deine erste Zeile durch die beiden Zeilen

```
import sys
```

```
f=open(sys.argv[1], "r")
```

ersetzt, ändert sich die Ausgabe wie folgt:

```

celinemueller@MacBook-Air-Buro sheet07 % python3 Umdrehen.py Umdrehen.py
['import sys\n', 'f=open(sys.argv[1], "r")\n', 'lines=f.readlines()\n', 'f.close()\n', 'print(lines)\n', '\n', 'for i in range(len(lines)-1,-1,-1):\n', '    #print(lines[i])\n', '    line=lines[i]\n', '    for j in range(len(line)-1,-1,-1):\n', '        print(line[j], end="")\n', '\n', '#i=0\n', '#while i< len(lines):\n', '    #    print(lines[-1-i], end="")\n', '#        i+=1']
1+=i    #
)""=dne ,li-1[senil(tnirp    #
:)senil(nel <i elihw#
0=i#

)""=dne ,lj[enil(tnirp
:)1-,1-,1-)enil(nel(egnar ni j rof
li[senil=enil
)]i[senil(tnirp#
:)1-,1-,1-)senil(nel(egnar ni i rof

)senil(tnirp
)(esolc.f
)(senildaer.f=senil
)"r" ,l1[vgra.sys(nepo=f
sys tropmi%
celinemueller@MacBook-Air-Buro sheet07 %

```

Diese Ausgabe ist quasi gleich zu der Master-Ausgabe, ausser dass du noch ein überflüssiges `print(lines)` Statement in Zeile 4 gemacht hast. Positiv hervorzuheben finde ich also den korrekten Gebrauch von zwei for-Schleifen, um rückwärts über alle Zeilen und alle Zeichen in jeder Zeile zu iterieren.

STARTING TASK Rechnen.py

THE ATTEMPTED SOLUTION IS

```

import sys

def minimum(zahlen):
    tmp=zahlen[0]
    for zahl in zahlen:
        if zahl<tmp:
            tmp=zahl
    return(tmp)

def max(zahlen):
    tmp=zahlen[0]
    for zahl in zahlen:

```

```

        if zahl>tmp:
            tmp=zahl
    return(tmp)

def durchschnitt(zahlen):
    summe=sum(zahlen)
    d=summe/len(zahlen)
    return(d)

zahlen=[]
for i in sys.stdin:
    zahlen.append(float(i))
#print(zahlen)

mini=minimum(zahlen)
print(mini)
maxi=max(zahlen)
print(maxi)
durchi=durchschnitt(zahlen)
print(durchi)

```

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
1.0
11.0
4.95
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.
