

# RUNNING TESTS FOR ASSIGNMENT SHEET06

STUDENT IS Goese

ATTEMPTED TASKS ARE Ausgabe.py  
Primzahlen.txt JedeZweite.py

STARTING TESTING Ausgabe.py

THE ATTEMPTED SOLUTION IS

```
def rekursion(i):  
    print("Rekursion")  
    if i < 10:  
        rekursion(i+1)  
  
rekursion(1)
```

THE MASTER SOLUTION IS

```
with open("Ausgabe.py", "r") as f:  
    for line in f.readlines():  
        print(line, end="")
```

THE INPUT DATA CONFIG FOR THIS FILE IS:

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

```
}  
}
```

## TEST RUN

### chosen input

"Ausgabe.py"

*das ist ein relativer Pfad der auf die Datei im aktuellen Ordner, also auf uns selbst, verweist. Dieser muss also existieren und nicht leer sein.*

### your output

```
celinemueller@MacBook-Air-Buro sheet06 % python3 Ausgabe.py  
Rekursion  
Rekursion  
Rekursion  
Rekursion  
Rekursion  
Rekursion  
Rekursion  
Rekursion  
Rekursion  
Rekursion
```

### expected output

```
celinemueller@Air-Buro sheet06 % python3 Ausgabe.py  
with open("Ausgabe.py", "r") as f:  
    for line in f.readlines():  
        print(line,end="")%
```

## COMMENTS ON YOUR SOLUTION:

- Du hast anscheinend die falsche Ausgabe bearbeitet... deine Lösung ist eine Funktion, die sich selbst 10 Mal aufruft, nicht ein Programm, das sich selbst bei Ausführung ausgibt.

---

## STARTING TESTING Primzahlen.txt

### THE ATTEMPTED SOLUTION IS

```
1 2
2 3
3 5
4 7
5 11
6 13
# ...
495 3539
496 3541
497 3547
498 3557
499 3559
500 3571
```

### THE MASTER SOLUTION IS

```
2
3
5
7
11
13
# ...
3539
3541
3547
3557
```

3559

3571

## THE INPUT DATA CONFIG FOR THIS FILE IS:

```
{  
    "compare_with": "Primzahlen_Master.txt"  
}
```

## TEST RUN

txt submission are not run.

## COMMENTS ON YOUR SOLUTION:

Du hast die richtigen Primzahlen ausgegeben, allerdings war in der Aufgabenstellung explizit gefordert, dass in einer Zeile "nur eine Primzahl" steht und sonst nichts.

---

## STARTING TESTING JedeZweite.py

## THE ATTEMPTED SOLUTION IS

```
import sys  
  
zeilen=sys.stdin.readlines()  
  
for i in range(0,len(zeilen),2):  
    print(zeilen[i])
```

# THE MASTER SOLUTION IS

```
import sys

a = sys.stdin.readlines()

for i in range(1, len(a), 2):
    print(a[i], end="")
```

## THE INPUT DATA CONFIG FOR THIS FILE IS:

```
{
  "stream": [
    {
      "type": "str",
      "repeatable": true,
      "min_repeats": 1,
      "max_repeats": 20,
      "requires_eof": true
    }
  ]
}
```

## TEST RUN

### chosen input

"|  
input  
a  
stream  
end  
with

ctrl-d"

*das sind Strings, die ich über die Standardeingabe eingeben kann.*

## your output

```
celinemueller@MacBook-Air-Buro sheet06 % python3 JedeZweite.py
I
input
a
stream
end
with
ctrl-d
I

a

end

ctrl-d

celinemueller@MacBook-Air-Buro sheet06 %
```

## expected output

```
celinemueller@Air-Buro sheet06 % python3 JedeZweite.py
I
input
a
stream
end
with
ctrl
d
input
stream
with
d
celinemueller@Air-Buro sheet06 %
```

## COMMENTS ON YOUR SOLUTION:

- Du hast den Eingabestrom mit dem `sys.stdin` - Modul eingelesen +1
  - Du hast jede zweite Zeile des Eingabestroms wieder ausgegeben +1
  - `readlines()` fügt am Ende jeder Zeile (jedes Strings) ein `"\n"` ein. `print()` fügt aber auch am Ende jeder Ausgabe ein `"\n"` ein, die eingelesenen Zeilen werden also mit doppeltem Zeilenumbruch wieder ausgegeben. Vergleiche nochmal deine und die Musterlösung, und finde den kleinen Unterschied. -1
-