

RUNNING TESTS FOR ASSIGNMENT SHEET06

STUDENT IS Demirel

ATTEMPTED TASKS ARE Ausgabe.py
JedeZweite.py

STARTING TESTING Ausgabe.py

THE ATTEMPTED SOLUTION IS

```
with open(__file__, 'r') as file:
    source_code = file.read()
    print(source_code)
```

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@Air-Buro sheet06 % python3 Ausgabe.py
with open(__file__, 'r') as file:
    source_code = file.read()
    print(source_code)
```

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:

1. Du hast `__file__` anstatt eines relativen oder absoluten Pfades auf die Datei `Ausgabe.txt` verwendet. +1
2. du hast `.read()` benutzt, welches die eingelesenen Zeilen direkt wieder ausgeben kann, ohne dass die Formatierung geändert wird +1

STARTING TESTING Primzahlen.txt

THE ATTEMPTED SOLUTION IS

```
new_list = []
with open ("jzText", "r") as q:
    new_list = q.readlines()
counter = 1
for line in new_list:
    if counter % 2 != 0:
        print(line)
    counter += 1
```

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,
      "need_eof": true
    }
  ]
}
```

TEST RUN

chosen input

"This"

"is"

"a"

"test"

"stream"

"<Ctrl-d>"

<Ctrl-d> ist kein utf-8 Zeichen, ich möchte nur darauf hinweisen, dass bei nicht-beendeter Eingabe auch das Programm nicht endet, also ein EOF notwendig ist.

your output

```
celinemueller@Air-Buro sheet06 % python3 JedeZweite.py
Traceback (most recent call last):
  File "/Users/celinemueller/Desktop/codeTestEnvironment/Handins/Demirel/sheet06/JedeZweite.py", line 2, in <module>
    with open ("jzText", "r") as q:
           ~~~~~^~~~~~
FileNotFoundError: [Errno 2] No such file or directory: 'jzText'
celinemueller@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 per relativem Dateipfad eine lokale Datei geöffnet und deren Zeilen in eine Liste gelesen, anstatt das Einlesen des (während runtime erzeugten) Input-Streams zu ermöglichen -1
- Anm.: technisch gesehen liest du so auch von der Standardeingabe ein:

```
in_char = input()
while in_char != "":
    in_char = input()
```

... da input() von der Standardeingabe einliest. Eleganter und vermutlich effizienter wäre es, die von Python bereitgestellte `sys.stdin.***`

() Funktionen zu verwenden, siehe zB Musterlösung.

- Es wird mit der ersten statt der zweiten Zeile anfangen zu printen (nutze lieber `if counter % 2 == 0: print(line)`) -1
- Anmerkung: das manuelle Setzen des Counters ist ein bisschen unnatürlich. denn wir durchlaufen danach sowieso eine schleife mithilfe eines Counters (der Iterationsvariablen, der Counter ist hier also soz. doppelt). Das ginge zB mit `for i in range(len(new_list)): [...]` `print(new_list[i])` Wir setzen manchmal extra Counter, um mathematische Operationen zu erleichtern etc (ich denke das wurde hier versucht). Aber die Anweisung "Jede Zweite Zeile" lässt sich aber vermutlich leichter umsetzen. Da `new_list` eine endliche Liste ist, können wir zB mit der Länge der Liste arbeiten und in Zweierschritten über `new_list` iterieren, wie in der Musterlösung . Aber so wie du es gemacht hast, funktioniert es auch .

STARTING TESTING JedeZweite.py

INFORMATION

The expected hand-in was a file containing only the primes, vertically listed
Das Programm, das du eingereicht hast, erzeugt den richtigen Output.
Man kann Performance-Steigerungen in der Berechnung der Primzahlen
anwenden, siehe dazu die extra-Datei.