

```
dollarWords10_20_TuesdayHASbug.py - C:\Python34\dollarWords10_20_TuesdayHASbug.py (3.4.3)
File Edit Format Run Options Window Help
fin = open("dictionary.txt", "r")
fout = open("dollarWordsOct20thWithNewline.txt", "w")
for original in fin:
    cost = 0
    original = original[:-1] # remove the line feed character, the '\n' new line char.
    for ch in original:
        if ch.isalpha():
            asciiValue = ord( ch.lower() )
            pennyValue = asciiValue - 96
            cost = cost + pennyValue
    if cost == 100:
        fout.write(original + "\n") # VIP: put back the '\n' new line character!!!
        # if cost == 100:
        #     fout.write(original) <-- PUTS ALL WORDS
        #                                     ON SAME 1st LINE!
fin.close()
fout.close()
print("The file 'dollarWordsOct20thWithNewline.txt' has been created.")
Ln: 19 Col: 45
```

```
dollarWordsOct20th.txt - C:\Python34\dollarWordsOct20th.txt (3.4.3)
File Edit Format Run Options Window Help
abettorsacolytesaequorinagrologyairdropsalertestalpinistamortiseamputeesamusedly
Ln: 1 Col: 48
```

Line #1 Column #48 Cursor is between **alertest** and **alpinist** words, both dollar words.

Ln: 1 Col 48

```
dollarWordsOct20th.txt - C:\Python34\dollarWordsOct20th.txt (3.4.3)
File Edit Format Run Options Window Help
arrowyeastingyelpersyirringytterbiazaptiahzaratitezestingzinkifyzithernzoogleas
Ln: 1 Col: 7607
```

Line #1 Column #7607 Cursor is between **zaratite** and **zesting** words, the 5th and 4th to last words! **Ln 1 Col 7607**

Here is the output of the wc (word count or Word Count = wc) Unix command for the output file with all of the 1,018 words on the same line in the first file, instead of being one word per line!

```
windsor:~/web/1510/files> wc dollarWordsOct20th.txt
0 1 7636 dollarWordsOct20th.txt
```

```
windsor:~/web/1510/files> wc dollarWordsOct20thWithNewline.txt
1018 1018 8654 dollarWordsOct20thWithNewline.txt
```

Look at and study the numbers: Notice that the first file has 7,636 bytes. There are 1,018 words. If we put an invisible NEWLINE character '\n' back at the end of each word, we will have 7636 + 1018 = 8654 bytes or characters in the file!

```
windsor:~/web/1510/files> wc dollarWordsOct20th.txt
```

```
0    1 7636 dollarWordsOct20th.txt
```

```
windsor:~/web/1510/files> wc dollarWordsOct20thWithNewline.txt
```

```
1018 1018 8654 dollarWordsOct20thWithNewline.txt
```

```
windsor:~/web/1510/files> bc                ← bc is the basic calculator on Unix
7636 + 1018
8654
```

```
windsor:~/web/1510/files> man wc
```

NAME

wc - print **newline**, **word**, and **byte** counts for each file

So notice that the file dollarWordsOct20th.txt has 0 newlines, 1 word, and 7636 bytes!

On the other hand, the file dollarWordsOct20thWithNewline.txt has 1018 newlines, 1018 words, and 8654 bytes!

Here are the first 11 lines of the file and the last 8 lines of the file of DOLLAR WORDS. Showing where the cursor was in the screen snapshots above of the one word and one line file, between 6th and 7th word and between 5th and 4th to last words.

```
abettors
acolytes
aequorin
agrology
airdrops
alertest
```

10/20/2015 6:07 PM

alpinist

```
amortise
amputees
amusedly
analysis
```

...

```
yirring
ytterbia
zaptiahs
zaratite
```

zesting

```
zinkify
zithern
zoogleas
```

fout.write(original) causes the problem because we took off the NEWLINE character.

fout.write(original + "\n") solves the problem!