

```

File: dollarWords
Author: CS 1510
Description: Uses a file dictionary to compute dollar words
based on penny math. Opens a dictionary file and reads it line
by line. Computes value and writes word to output file if it
is worth exactly 100 cents.

```

```

# Open our files
fin = open("dictionary.txt", "r")
fout = open("extracredit.txt", "w")

```

#For every line, get each word

```

firstLetter = 'a'
firstCount = 0
dollarWords = 0
words = 0

```

for original in fin:

```

original = original[:-1] #Why this line?
cost = 0

```

#For every word, compute the penny math

```

for char in original:
    value = ord(char)
    if char>="a" and char<="z":
        cost = cost+(value-96)
    elif char>="A" and char<="Z":
        cost = cost+(value-64)

```

```
words = words + 1
```

#Write dollar words to file

```

if cost==100:
    dollarWords = dollarWords + 1
    if original[0] != firstLetter:
        fout.write("\n *** " + firstLetter.upper() + firstLetter \
            + " " + str(firstCount) + " entries.\n\n")

```

```

print(firstLetter.upper() + firstLetter + " " \
    + str(firstCount) + " dollar words.")

```

```

firstLetter = original[0]
firstCount = 0

```

```
firstCount += 1
```

```
fout.write(original + " ---> ")
```

for ch in original:

```
if ch in "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ":
```

```
chCost = ord( ch.lower() ) - 96
```

```
fout.write(ch + " " + str(chCost) + " ")
```

```
fout.write("\n")
```

```

fout.write("\n *** " + firstLetter.upper() + firstLetter + " " \
    + str(firstCount) + " entries.\n\n")

```

```

print(firstLetter.upper() + firstLetter + " " + str(firstCount) \
    + " dollar words.")

```

print("\n\nOutput file written called 'extracredit.txt'")

```

print("\nThere were " + str(dollarWords) + " dollar words out of " \
    + str(words) + " dictionary words.")

```

```

fin.close()
fout.close()

```

```

abettors ----> a 1 b 2 e 5 t 20 t 20 o 15 r 18 s 19
acolytes ----> a 1 c 3 o 15 l 12 t 20 e 5 s 19
aequorin ----> a 1 e 5 q 17 u 21 o 15 r 18 i 9 n 14
agrology ----> a 1 g 7 r 18 o 15 l 12 o 15 g 7
airdrops ----> a 1 i 9 r 18 d 4 r 18 o 15 p 16 s 19
alertest ----> a 1 l 12 e 5 r 18 t 20 e 5 s 19 t 20
alpinist ----> a 1 l 12 p 16 i 9 n 14 i 9 s 19 t 20
amortise ----> a 1 m 13 o 15 r 18 t 20 i 9 s 19 e 5
amputees ----> a 1 m 13 p 16 u 21 t 20 e 5 e 5 s 19
amusedly ----> a 1 m 13 u 21 s 19 e 5 d 4 l 12
analysis ----> a 1 n 14 a 1 l 12 s 19 i 9 s 19
aneurism ----> a 1 n 14 e 5 u 21 r 18 i 9 s 19 m 13
anginous ----> a 1 n 14 g 7 i 9 n 14 o 15 u 21 s 19
annually ----> a 1 n 14 n 14 u 21 a 1 l 12 l 12
anterior ----> a 1 n 14 t 20 e 5 r 18 i 9 o 15 r 18
anticult ----> a 1 n 14 t 20 i 9 c 3 u 21 l 12 t 20
antislip ----> a 1 n 14 t 20 i 9 s 19 l 12 i 9 p 16
apiology ----> a 1 p 16 i 9 o 15 l 12 o 15 g 7
apostacy ----> a 1 p 16 o 15 s 19 t 20 a 1 c 3
applying ----> a 1 p 16 p 16 l 12 i 9 n 14 g 7
apropos ----> a 1 p 16 r 18 o 15 p 16 o 15 s 19
arrests ----> a 1 r 18 r 18 e 5 s 19 t 20 s 19
arrivals ----> a 1 r 18 r 18 i 9 v 22 a 1 l 12 s 19
arrowy ----> a 1 r 18 r 18 o 15 w 23
arugulas ----> a 1 r 18 u 21 g 7 u 21 l 12 a 1 s 19
asbestos ----> a 1 s 19 b 2 e 5 s 19 t 20 o 15 s 19
atomiser ----> a 1 t 20 o 15 m 13 i 9 s 19 e 5 r 18
atonally ----> a 1 t 20 o 15 n 14 a 1 l 12 l 12
attitude ----> a 1 t 20 t 20 i 9 t 20 u 21 d 4 e 5
attunes ----> a 1 t 20 t 20 u 21 n 14 e 5 s 19
augmenta ----> a 1 u 21 g 7 m 13 e 5 n 14 t 20 s 19
auntlier ----> a 1 u 21 n 14 t 20 l 12 i 9 e 5 r 18
awfully ----> a 1 w 23 f 6 u 21 l 12 l 12
azurite ----> a 1 z 26 u 21 r 18 i 9 t 20 e 5

```

*** Aa 34 entries.

```

bulliest ----> b 2 u 21 l 12 l 12 i 9 e 5 s 19 t 20
bullnose ----> b 2 u 21 l 12 l 12 n 14 o 15 s 19 e 5
byplays ----> b 2 p 16 l 12 a 1 s 19
*** Bb 35 entries.
calypter ----> c 3 a 1 l 12 p 16 t 20 e 5 r 18

```