CSC 242: Introduction to Computer Science II

Assignment 5

Due Monday, May 4th, at NOON

Reading

Review Chapter 8 and read Chapter 9 in **Introduction to Computing using Python: An Application Development Focus, Second Edition** by Ljubomir Perković.

Logistics

In this class programming assignments may be completed in consultation with up to two other classmates. You must identify the classmates with whom you collaborate in a comment at the top of the assignment, and the number of collaborators on any assignment **may not exceed two other people**. You must also submit a comment in your submission for each assignment that describes in detail how each collaborator contributed to the assignment. If you did not collaborate with anyone on the assignment, you must include a comment that says that. You may not under any circumstances discuss the assignments with classmates' other than your identified collaborators. Working so closely with anyone other than your identified collaborators, Mr. Zoko, or lab assistant, so as to produce identical or near identical code is a violation of the Academic Integrity policy. This policy will be strictly enforced.

Please include the following with your assignment submission:

1. A comment at the top of your Python file identifying any classmates with whom you discussed or in any other way collaborated on the assignment. You may work (directly or indirectly) with **no more than two** other people.
2. Add a comment at the top of your Python file that describes for each person what they contributed to the assignment. This must be at least 2-3 sentences and be **very specific and detailed**.

A submission that does not include a list of collaborators and a comment indicating how you collaborated with classmates will earn a 0. If you worked alone you must put a comment at the top of your file that indicates that or you will also receive a 0. There will be no exceptions to this rule.

Again, you are subject to all the rules specified in the Academic Integrity policy. Please read it carefully before beginning this assignment.

Assignment

Begin the assignment by downloading the **csc242hw5.py** file from the D2L site. It contains a significant amount of code to get you started with the GUI classes required for this assignment.

Submitting the assignment

You must submit the assignment using the assignment 5 dropbox on the D2L site. Submit a Python file (csc242hw45py) with your implementation in it and comments describing your collaboration status. Submissions after the deadline listed above will be automatically rejected by the system. See the syllabus for the grading policy.

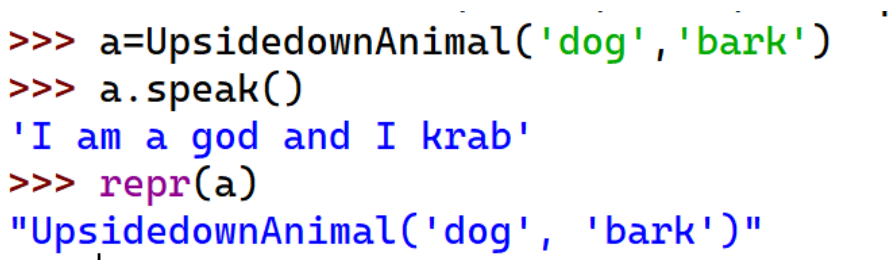
Grading

The assignment is worth 100 points. Any student who does not submit comments in the Python file describing the contributions of each team member or indicating that he/she worked alone will earn a 0 on the assignment.

PROBLEM 1 (20 POINTS):

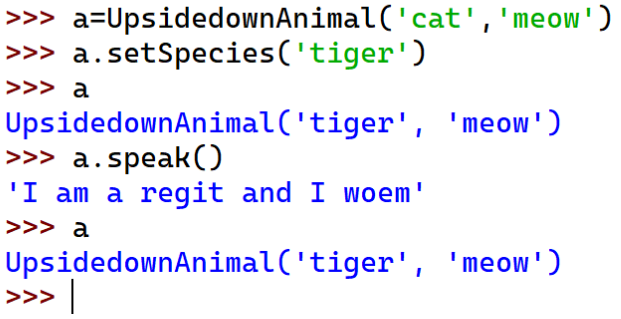
You must implement a class named UpsidedownAnimal that inherits from Animal. An Upsidedown animal looks like a normal Animal but it behaves a bit differently than a normal animal. (I've watched quite a few episodes of Stranger Things with my children lately for those wonder why it's called an UpsidedownAnimal.) No template is provided for you. You must implement the behavior as shown below using **NO MORE THAN TWO METHODS**. Hint: You need to override methods from Animal. You cannot modify Animal.

Sample 1:



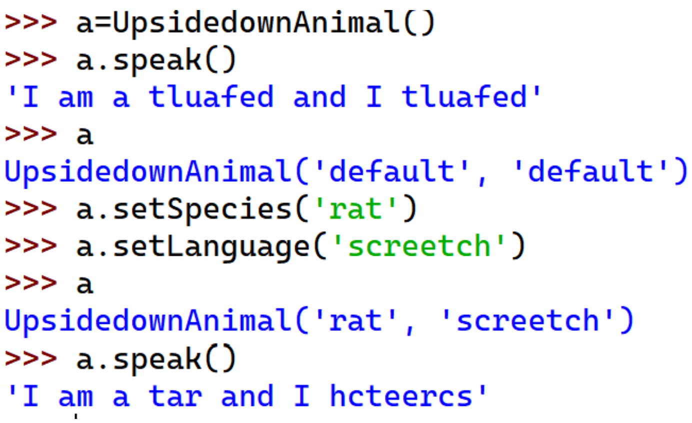
```
>>> a=UpsidedownAnimal('dog','bark')
>>> a.speak()
'I am a god and I krab'
>>> repr(a)
"UpsidedownAnimal('dog', 'bark')"
```

Sample 2:



```
>>> a=UpsidedownAnimal('cat','meow')
>>> a.setSpecies('tiger')
>>> a
UpsidedownAnimal('tiger', 'meow')
>>> a.speak()
'I am a regit and I woem'
>>> a
UpsidedownAnimal('tiger', 'meow')
>>> 
```

Sample 3:



```
>>> a=UpsidedownAnimal()
>>> a.speak()
'I am a tluafed and I tluafed'
>>> a
UpsidedownAnimal('default', 'default')
>>> a.setSpecies('rat')
>>> a.setLanguage('screetch')
>>> a
UpsidedownAnimal('rat', 'screetch')
>>> a.speak()
'I am a tar and I hcteercs'
```

PROBLEM 2 (40 POINTS):

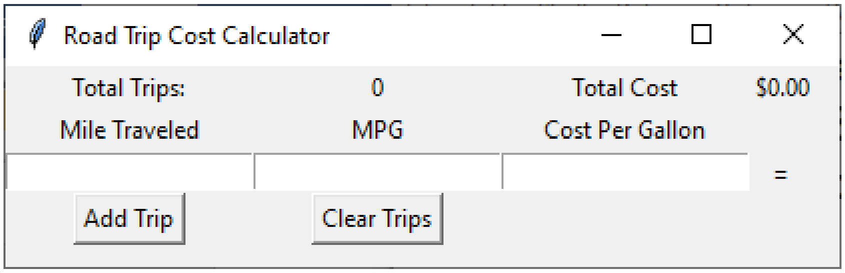
Implement the RoadTripCalculator GUI per the screenshots and docstrings found in the template file.

The RoadTripCalculator calculates the cost of a leg of a trip and tracks the combined cost off all the parts of a trip.

Every field must be validated and convertible to a float. App cannot crash -15 automatically deducted.

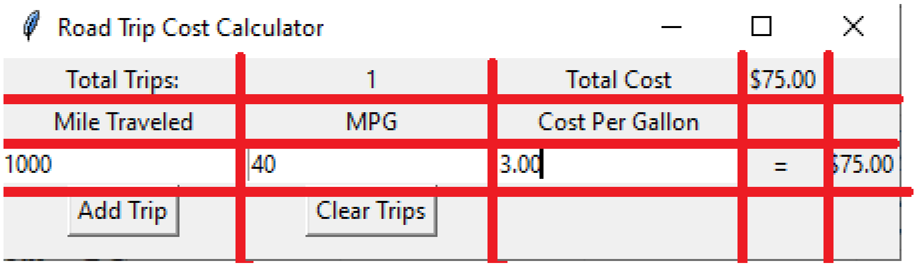
Grading Rubric:

- 20: GUI Matches screenshots
- 10: Validation messages match screenshots
- 10: Correct Calculation
- 10: Correct Tracking of total trips and total cost
- 10: Clear Button works as shown in screenshots



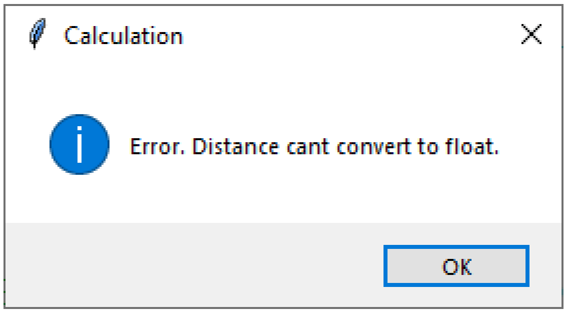
Total Trips:	0	Total Cost	$0.00	
Mile Traveled	MPG	Cost Per Gallon		
			=	
Add Trip	Clear Trips			

GUI HINT. It is a 4x5 grid:

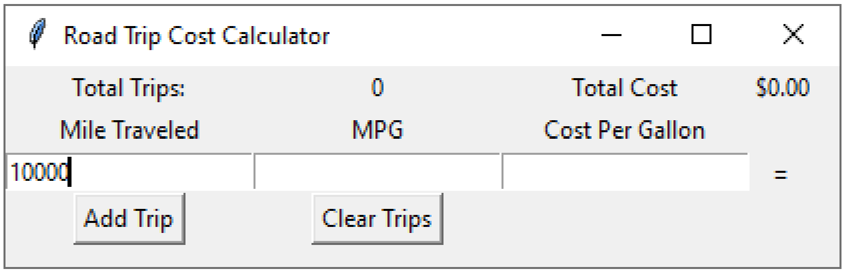


Total Trips:	1	Total Cost	$75.00	
Mile Traveled	MPG	Cost Per Gallon		
1000	40	3.00	=	$75.00
Add Trip	Clear Trips			

Click 'Add Trip'

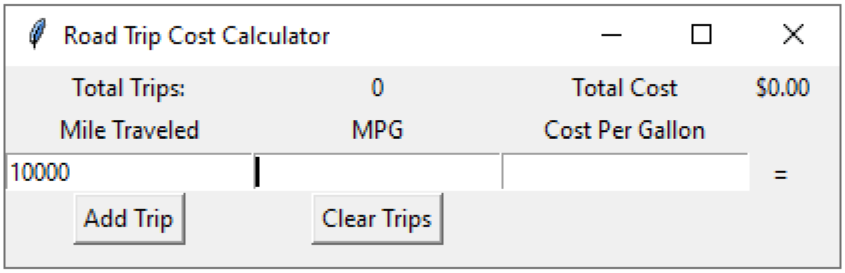


Error. Distance cant convert to float.

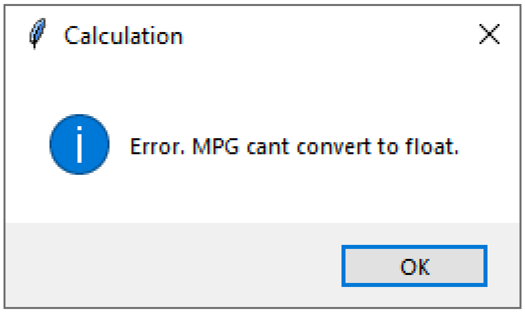


Mile Traveled: 10000

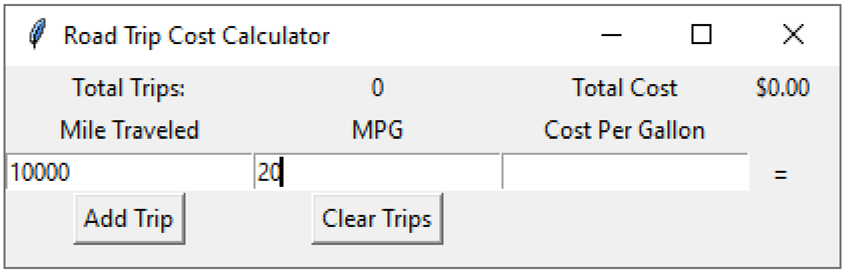
Click ‘Add Trip’



Mile Traveled: 10000



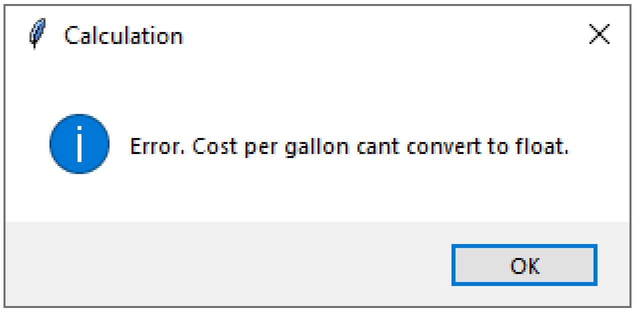
Error. MPG cant convert to float.



Mile Traveled: 10000

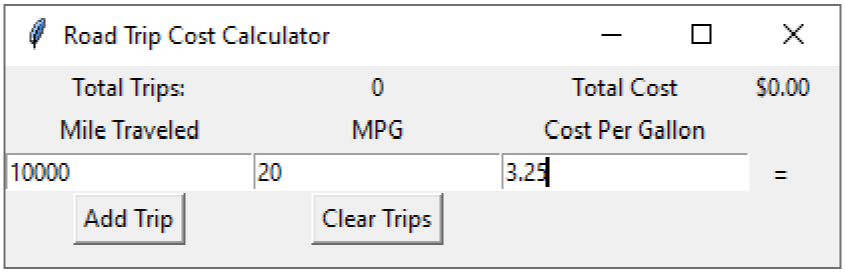
MPG: 20

Click ‘Add Trip’


Calculation

Error. Cost per gallon cant convert to float.

All three fields behave the same way

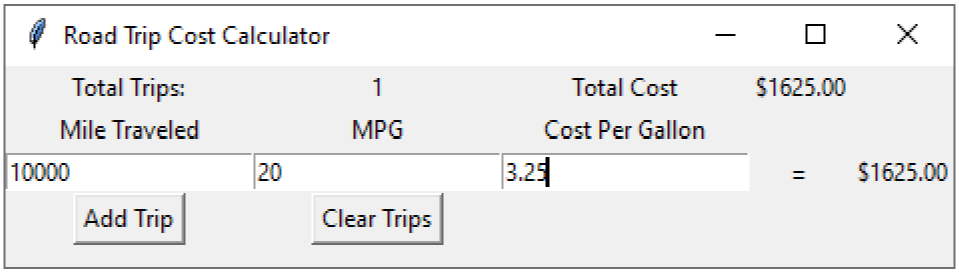

Road Trip Cost Calculator

Total Trips: 0 Total Cost $0.00

Mile Traveled MPG Cost Per Gallon

10000 20 3.25 =

Click ‘Add Trip’


Road Trip Cost Calculator

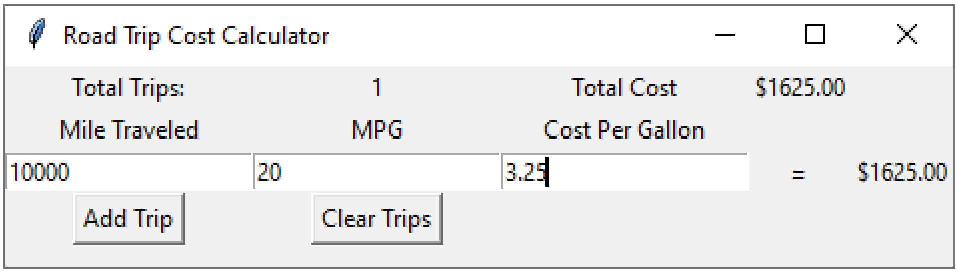
Total Trips: 1 Total Cost $1625.00

Mile Traveled MPG Cost Per Gallon

10000 20 3.25 = $1625.00

Notice the label with the trip calculation changes and the Total Trips and Total Cost increases.

Add another trip:

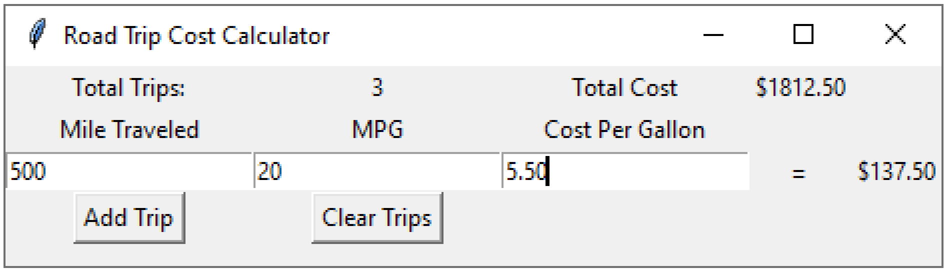

Road Trip Cost Calculator

Total Trips: 1 Total Cost $1625.00

Mile Traveled MPG Cost Per Gallon

10000 20 3.25 = $1625.00

Add another trip:



Total Trips: 3 Total Cost $1812.50

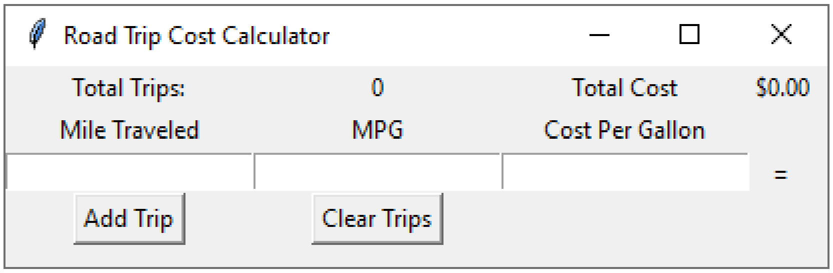
Mile Traveled: 500

MPG: 20

Cost Per Gallon: 5.50

= $137.50

Finally. Click “Clear Trips” and everything resets.



Total Trips: 0 Total Cost $0.00

Mile Traveled

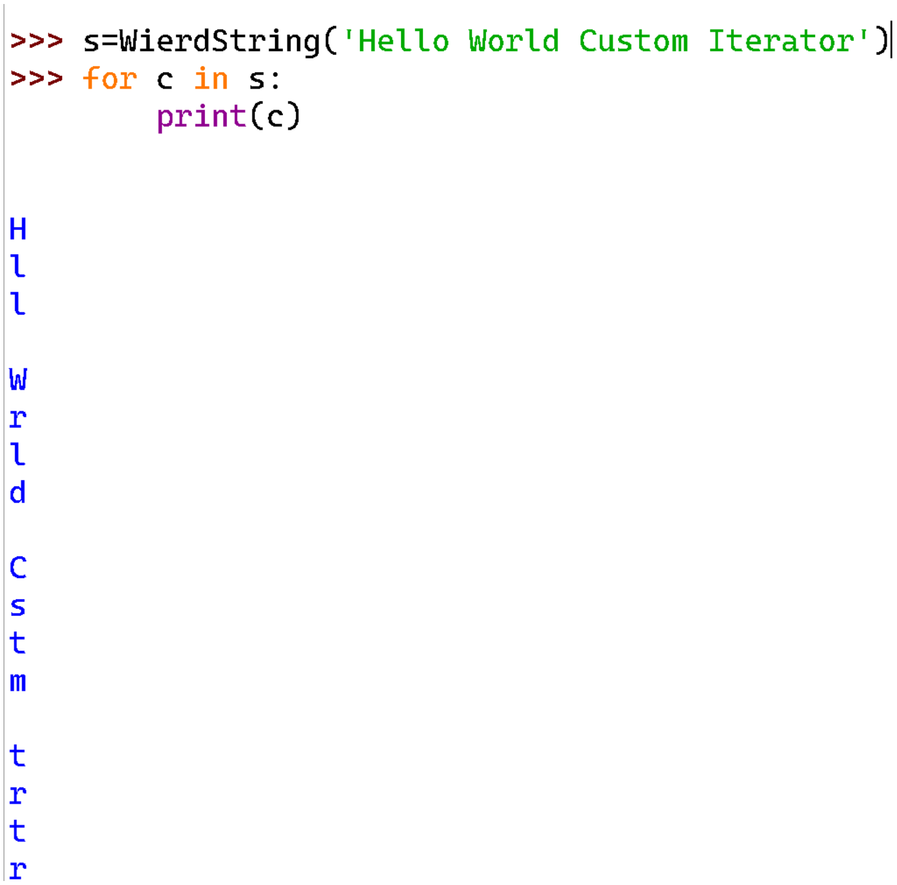
MPG

Cost Per Gallon

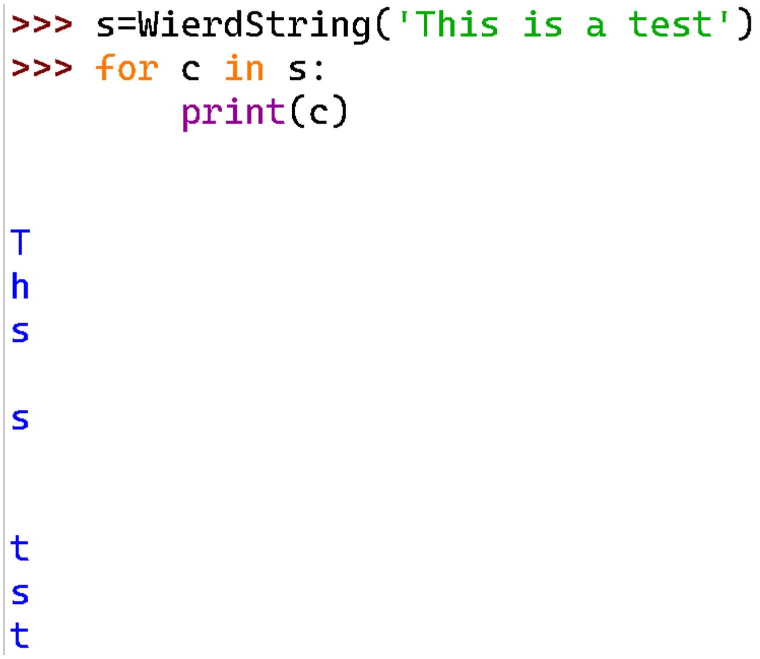
=

PROBLEM 3 (40 POINTS):

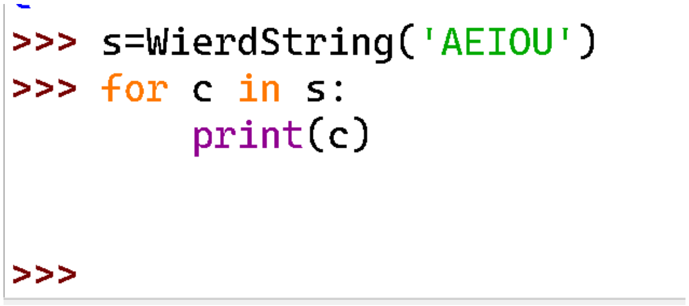
Implement the classes WeirdString and WierdStringIter. WierdString returns WierdStringIter which skips over vowels (upper and lower case) in a string. See screenshot below for behavior:



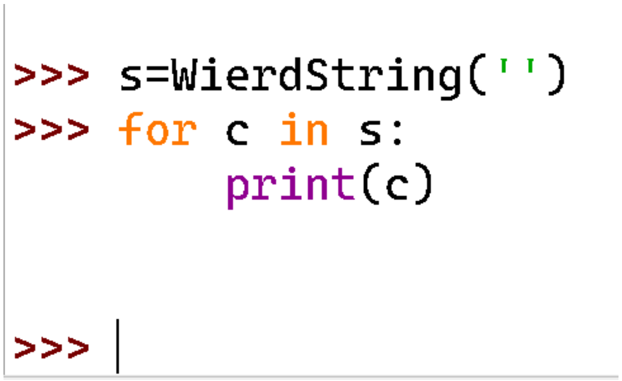
```
>>> s=WierdString('Hello World Custom Iterator')
>>> for c in s:
	print(c)


H
l
l

W
r
l
d

C
s
t
m

t
r
t
r
```



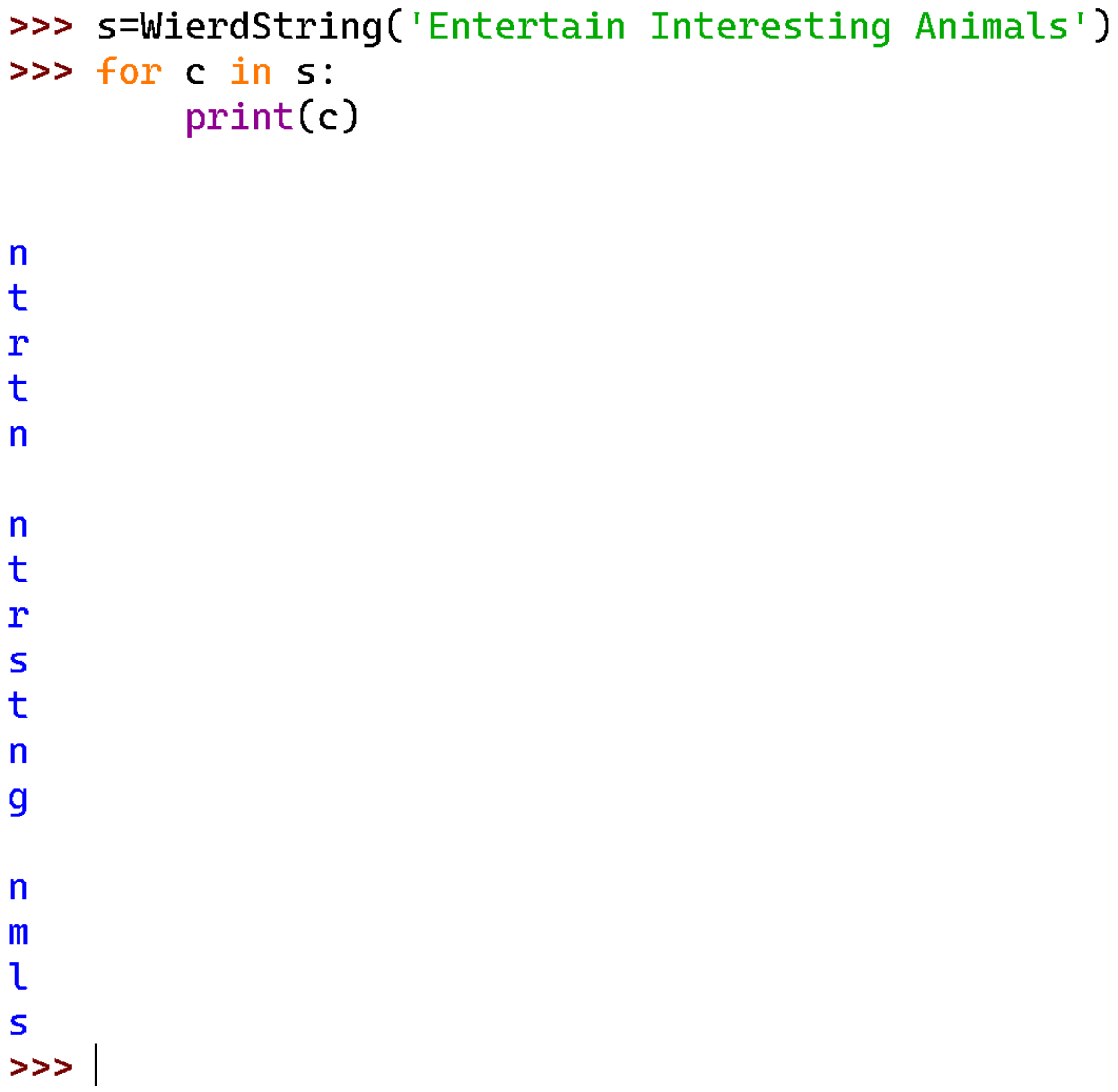
```
>>> s=WierdString('This is a test')
>>> for c in s:
        print(c)


T
h
s

s


t
s
t
```



```
>>> s=WierdString('AEIOU')
>>> for c in s:
        print(c)


>>> 
```



```
>>> s=WierdString('')
>>> for c in s:
        print(c)


>>> 
```



```
>>> s=WierdString('Entertain Interesting Animals')
>>> for c in s:
        print(c)


n
t
r
t
n
 
n
t
r
s
t
n
g
 
n
m
l
s
>>> 
```