Noble Menchaca
LING 388
Assignment 2

```
import nltk, re
from nltk.corpus import wordnet as wn
print(':: Assignment 2\n:: Noble Menchaca')
```

1. Wordnet, Hypernyms, and Semantic Similarity (3 pts)

(a) Find the lowest common hypernyms between each of these three animals: giant panda, dog, and tortoise.

```
panda = wn.synset('giant_panda.n.01')
dog = wn.synset('dog.n.01')
tort = wn.synset('tortoise.n.01')

print(panda.lowest_common_hypernyms(dog))
print(panda.lowest_common_hypernyms(tort))
print(dog.lowest_common_hypernyms(tort))
```

```
[Synset('carnivore.n.01')]
[Synset('vertebrate.n.01')]
[Synset('vertebrate.n.01')]
```

(b) Find the minimum depth for the three lowest common hypernym synsets found in (a).

```
print(wn.synset('carnivore.n.01').min_depth())
print(wn.synset('vertebrate.n.01').min_depth())
print(wn.synset('vertebrate.n.01').min_depth())
11
8
8
```

(c) Find the minimum semantic similarity (0-1) between the synsets for each of the animals in (a).

```
print(panda.path_similarity(dog))
print(panda.path_similarity(tort))
print(dog.path_similarity(tort))
```

```
0.2
0.09090909090909091
0.1
```

Noble Menchaca
LING 388
Assignment 2
2. Wordnet, Hypernyms, and Definitions (4 pts)

Write a function supergloss(s) that takes a synset (s) as its input, and returns a string consisting of the concatenation of the lemmas and definition of s, and the lemmas and definitions of all the hypernyms of s.

```
	motorcar = wn.synset('motor_vehicle.n.01')

	def supergloss(syn):
		string = ''
		paths = syn.hypernym_paths()
		for item in paths[0]:
			deff = str(item.definition())
			lemma = item.lemma_names()
		string += str(lemma)
		string += ' : '
		string += deff
		string += '\n'
	return string

print(supergloss(motorcar))
```

```
['entity'] : that which is perceived or known or inferred to have its own distinct
existence (living or nonliving)
['physical_entity'] : an entity that has physical existence
['object', 'physical_object'] : a tangible and visible entity; an entity that can
cast a shadow
['whole', 'unit'] : an assemblage of parts that is regarded as a single entity
['artifact', 'artefact'] : a man-made object taken as a whole
['instrumentality', 'instrumentation'] : an artifact (or system of artifacts) that is
instrumental in accomplishing some end
['container'] : any object that can be used to hold things (especially a large metal
boxlike object of standardized dimensions that can be loaded from one form of
transport to another)
['wheeled_vehicle'] : a vehicle that moves on wheels and usually has a container for
transporting things or people
['self-propelled_vehicle'] : a wheeled vehicle that carries in itself a means of
propulsion
['motor_vehicle', 'automotive_vehicle'] : a self-propelled wheeled vehicle that does
not run on rails
```

```
print('__Part3__')
```

Noble Menchaca
LING 388
Assignment 2
3. Regular Expressions (3 pts)

Write regular expressions to match the following classes of strings :
(a) A single determiner (assume that a, an, and the are the only determiners). The first character must be case insensitive.
(b) An arithmetic expression using only integers, addition, and multiplication, such as 2*3+8, 10*53, 20+35, 20+10+3+5, 22+30*1+3*4, etc.

```
strings = [
    'A few days ago I found a new recipe for a delicious sounding dessert',
    'The recipe was for an Orange Ricotta Pound Cake',
    'I\'ve been wanting to try a baking recipe with ricotta',
    'And this was the perfect opportunity',
    'What a tasty treat for the weekend',
    'I checked my fridge and found all necessary ingredients',
    'Most times I\'m missing something small like vanilla',
    'after measuring out all my materials',
    'I began to mix together flours and fruit and cheese',
    'Soon enough I had this flavorful breakfast item to pair with my coffee']
strings2 = [
    'I\'m afraid I\'m not motivated enough to write up excuses to use 
mathematical operations',
    'Instead I will say that I\'ve been really enjoying cooking\\baking 
recently',
    'so much so that I\'ve been bookmarking food blogs from all around the 
web to try new things',
    'It has gotten to the point where I need to have nested folders to 
organize all the recipes I save',
    'some for Thai food, for Indian, Japanese, some for dessert... but mostly 
I love breakfast recipes.',
    'OKay back to business, here are math expressions 33+234 234+56 11+00001 
9cc9+090',
    'And some more expressions 99+1 99*999 math math math 987+123',
    'Seriously though I\'m thinking about what to make next 55+45 12+68',
    'I just bought a bunch of strawberries on sale 7575+2342 998***909',
    'I\'ll probably look up more recipes after I\'m done with this 9+1 1+msm0 
+234+23 *3254*2345 123 24 1'
]

print('\n\n__Determiner Finder__')
for i in range(len(strings)):
    tokens = nltk.word_tokenize(strings[i])
    determiners = [w for w in tokens if re.search('^(A|a)n*$', w) or 
re.search('^(T|t)he$', w)]
    print('determiners for string '+ str(i+1) + '    :  ' + str(determiners))

print('\n\n___Math Expressions__')
```

Noble Menchaca
LING 388
Assignment 2

```
for i in range(len(strings2)):
    tokens = nltk.word_tokenize(strings2[i])
    maths = [w for w in tokens if re.search('^[0-9]+(\+|\*)[0-9]+$', w)]
    print('math statements for string '+ str(i+1) + ':  ' + str(maths))
```

```
__Determiner Finder__
determiners for string 1    :  ['A', 'a', 'a']
determiners for string 2    :  ['The', 'an']
determiners for string 3    :  ['a']
determiners for string 4    :  ['the']
determiners for string 5    :  ['a', 'the']
determiners for string 6    :  []
determiners for string 7    :  []
determiners for string 8    :  []
determiners for string 9    :  []
determiners for string 10    :  []


___Math Expressions__
math statements for string 1:  []
math statements for string 2:  []
math statements for string 3:  []
math statements for string 4:  []
math statements for string 5:  []
math statements for string 6:  ['33+234', '234+56', '11+00001']
math statements for string 7:  ['99+1', '99*999', '987+123']
math statements for string 8:  ['55+45', '12+68']
math statements for string 9:  ['7575+2342']
math statements for string 10:  ['9+1']
```