CSC241 - HW3

Reading: Chapter 3 and 4

Instructions:

1. **Read the HW guidelines** posted in d2l.depaul.edu > admin.
2. Your solution file MUST be named **hw3.py**. In a comment, include the name(s) of any collaborators.
3. To receive **full credit**, the **names** of files, functions and the **output** must be ***exactly*** as indicated here.
4. **Test your code** by downloading the file **hw3TEST.py** in the same working folder and typing in the shell:

```
>>> import doctest
>>> doctest.testfile('hw3TEST.py')
TestResults(failed=0, attempted=23)
>>>
```

If you prefer, you can make the test run automatically by adding the following lines at the bottom of your hw2 file/modfule. Every time you run the module, the doctest will be performed.

```
if __name__=='__main__':
    import doctest
    print( doctest.testfile('hw3TEST.py'))
```

Problems:

1. Implement a function `printMultiples` that takes two positive integers n and m as parameters, and **prints**, all on one line, the first m multiples of n. You may assume that n and m will be positive (> 0). The information below shows how you would call the function `printMultiples` and what it would display for several examples:

```
>>> printMultiples(10,2)
10 20
>>> printMultiples(10,10)
10 20 30 40 50 60 70 80 90 100
>>> printMultiples(2,12)
2 4 6 8 10 12 14 16 18 20 22 24
>>> printMultiples(3,5)
3 6 9 12 15
>>> printMultiples(7,5)
7 14 21 28 35
>>>
```

2. Write a function `customSpam` that takes three parameters, a string containing a first name followed by a space followed by a last name, a string representing a dollar amount spelled out in words, and string representing an e-mail address. The function **prints** a customized spam solicitation to the person using their name, the dollar amount, and the e-mail address. The name should use a capitalized letter for the first and last name regardless of the capitalization of the string entered and the dollar amount should be in all caps with spaces between each letter. The following shows several examples:

```
>>> customSpam('Ambrose Bierce','one
thousand','iamgreat@xyz.com')
Dear Ambrose Bierce,
We would like to let you know about a great opportunity.
You can make O N E   T H O U S A N D dollars in just a few
short weeks!
This is a limited-time offer.
Please contact us at iamgreat@xyz.com for more information.
>>> customSpam('haruki murakami','two
million','callnow@gmail.com')
Dear Haruki Murakami,
We would like to let you know about a great opportunity.
You can make T W O   M I L L I O N dollars in just a few short
weeks!
This is a limited-time offer.
Please contact us at callnow@gmail.com for more information.
>>> customSpam('ORANGE juice','four
hundred','notimeleft@aol.com')
Dear Orange Juice,
```

```
We would like to let you know about a great opportunity.
You can make F O U R   H U N D R E D dollars in just a few
short weeks!
This is a limited-time offer.
Please contact us at notimeleft@aol.com for more information.
>>>
```

3. Implement a function `ion2e` that takes a string as a parameter. If the string ends with `'ion'` it **returns** the initial part of the string (before the `'ion'`) followed by an `'e'` with no extra spaces. If the string does not end with `'ion',` including the circumstance in which the string contains 'ion' as a substring, it **returns** the original string. The following shows several examples of how the function would be used:

```
>>> ion2e('congratulation')
'congratulate'
>>> ion2e('marathon')
'marathon'
>>> ion2e('accordionist')
'accordionist'
>>> ion2e('ionization')
'ionizate'
>>> ion2e('congratulation')=='congratulate'  # returns?
True
>>> x = ion2e('congratulate') # return
>>> x
'congratulate'
```

4. Write a function `startsWith` that takes two arguments, a target string and a list of words. It then prints each word in the list that starts with the target string. To receive full credit, the testing should not be case sensitive, it ignores the case of both the target and the word considered. (see the output when called with the target `'App'`). Sample output:

```
>>>
startsWith('a',['apple','ApPle','orange','Apple','kiwi','apric
ot'])
apple
ApPle
Apple
apricot
>>>
startsWith('A',['apple','ApPle','orange','Apple','kiwi','apric
ot'])
apple
ApPle
Apple
apricot
```

```
>>>
startsWith('App',['apple','ApPle','orange','Apple','kiwi','apr
icot'])
apple
ApPle
Apple
>>>
startsWith('orang',['apple','ApPle','orange','Apple','kiwi','a
pricot'])
orange
>>>
startsWith('ORANG',['apple','ApPle','orange','Apple','kiwi','a
pricot'])
orange
>>>
startsWith('an',['apple','ApPle','orange','Apple','kiwi','apri
cot'])
>>>
startsWith('ppl',['apple','ApPle','orange','Apple','kiwi','apr
icot'])
>>>
```