CSC 241-404: Introduction to Computer Science I

Assignment 3

Reading

Review Chapter 3 and read Chapter 4 in your text

Logistics

Please read the Academic Integrity pledge prior to beginning this assignment. It is crucial that you understand what kind of collaboration is allowed and what kind is disallowed on assignments. The Academic Integrity pledge can be found on the D2L site.

In this class programming assignments may be completed in consulation 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 include a comment in your submission for each assignment that describes in detail how each collaborator contribued 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 identifed collaborators, Dr. Settle, or Ms. Troost, 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 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 file describing for each person what they contributed to the assignment. This must be at least 2-3 sentences and be very specific and detailed.

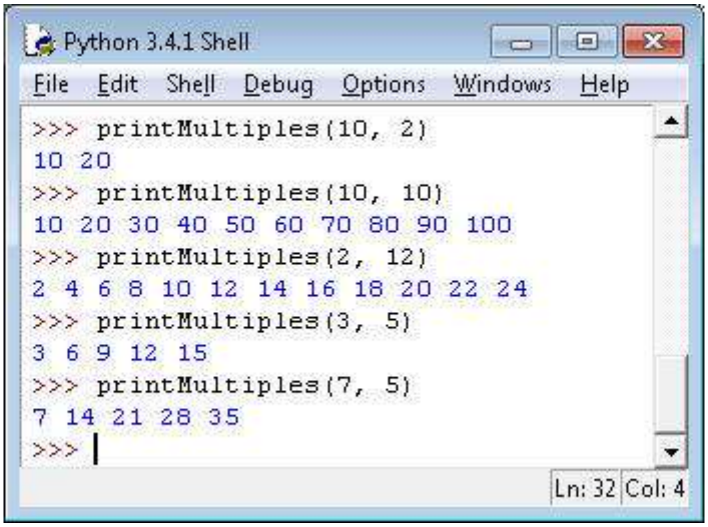
A submission that does not include a list of collaborators and comments indicating how you collaborated with classmates will earn a 0. If you worked alone you must put a comment at the top of your file that indicates that or you will also receive a 0.

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

Assignment

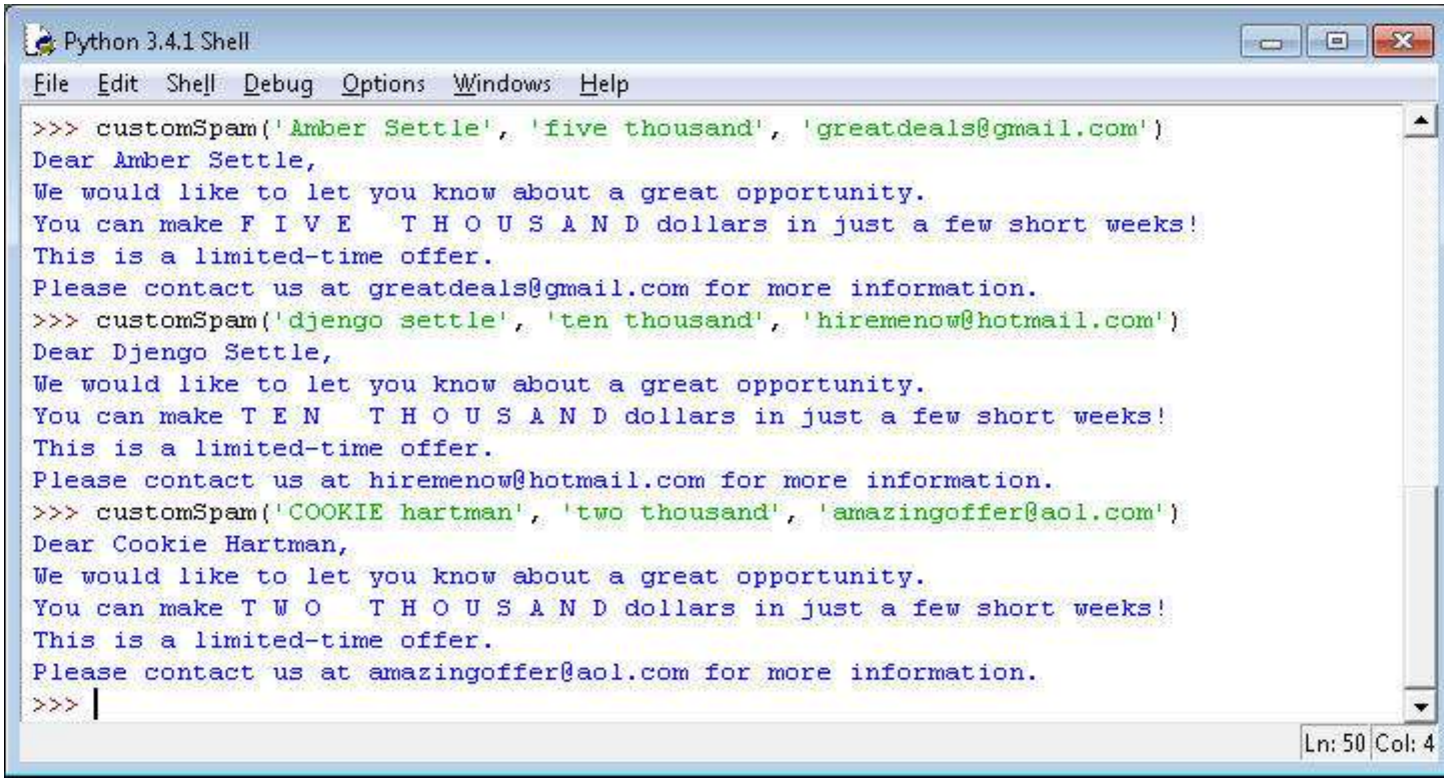
Implement the functions below in the file **csc241hw**3**.py** which can be found on the D2L site in the third assignment section. You should save the template file I provided and then modify that file by adding the bodies for the functions. When you do, make sure to remove the placeholder pass statements that are currently there. **You must also write doc strings for every function**. A submission without doc strings will not earn full credit.

1. Implement a function **printMultiples**() that takes two positive integers n and m as parameters, and **prints**, all on one line, all 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



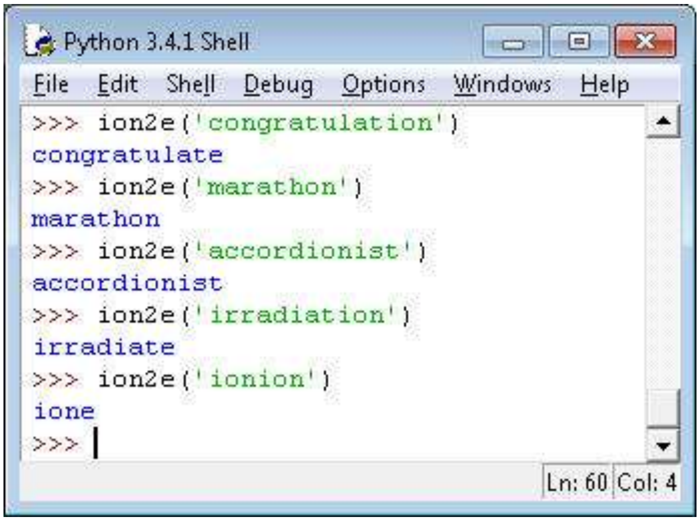
```
>>> printMultiples(10, 2)
10 20
>>> printMultiples(10, 10)
10 20 30 40 50 60 70 80 90 100
>>> printMultiples(2, 12)
2 4 6 8 10 12 14 16 18 20 22 24
>>> printMultiples(3, 5)
3 6 9 12 15
>>> printMultiples(7, 5)
7 14 21 28 35
>>> 
```

2. Write a function **customSpam**() that takes as 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 and **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('Amber Settle', 'five thousand', 'greatdeals@gmail.com')
Dear Amber Settle,
We would like to let you know about a great opportunity.
You can make F I V 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 greatdeals@gmail.com for more information.
>>> customSpam('djengo settle', 'ten thousand', 'hiremenow@hotmail.com')
Dear Djengo Settle,
We would like to let you know about a great opportunity.
You can make T E N   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 hiremenow@hotmail.com for more information.
>>> customSpam('COOKIE hartman', 'two thousand', 'amazingoffer@aol.com')
Dear Cookie Hartman,
We would like to let you know about a great opportunity.
You can make T W O   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 amazingoffer@aol.com for more information.
>>> 
```

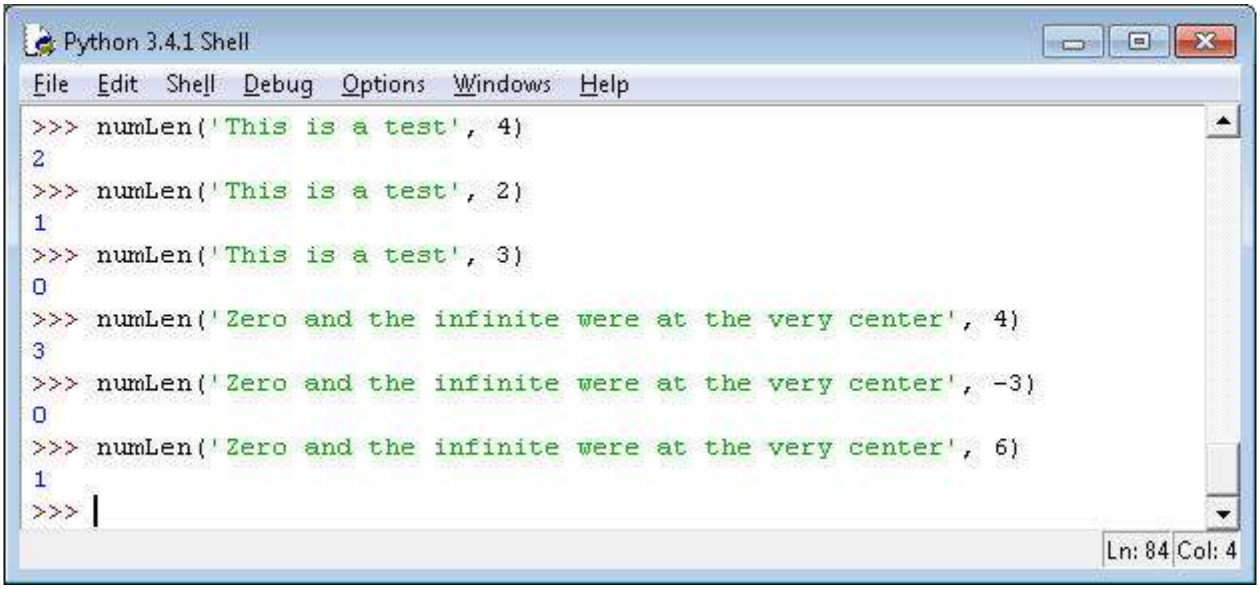
3. Implement a function **ion2e**() that takes a string as a parameter. If the string ends with 'ion' it **prints** the initial part of the string (before the 'ion') followed by an 'e' with no extra spaces. If the string does not end with 'ion', including the circumstance in which the string contains 'ion' as a substring, it **prints** 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('irradiation')
irradiate
>>> ion2e('ionion')
ione
>>> 
```

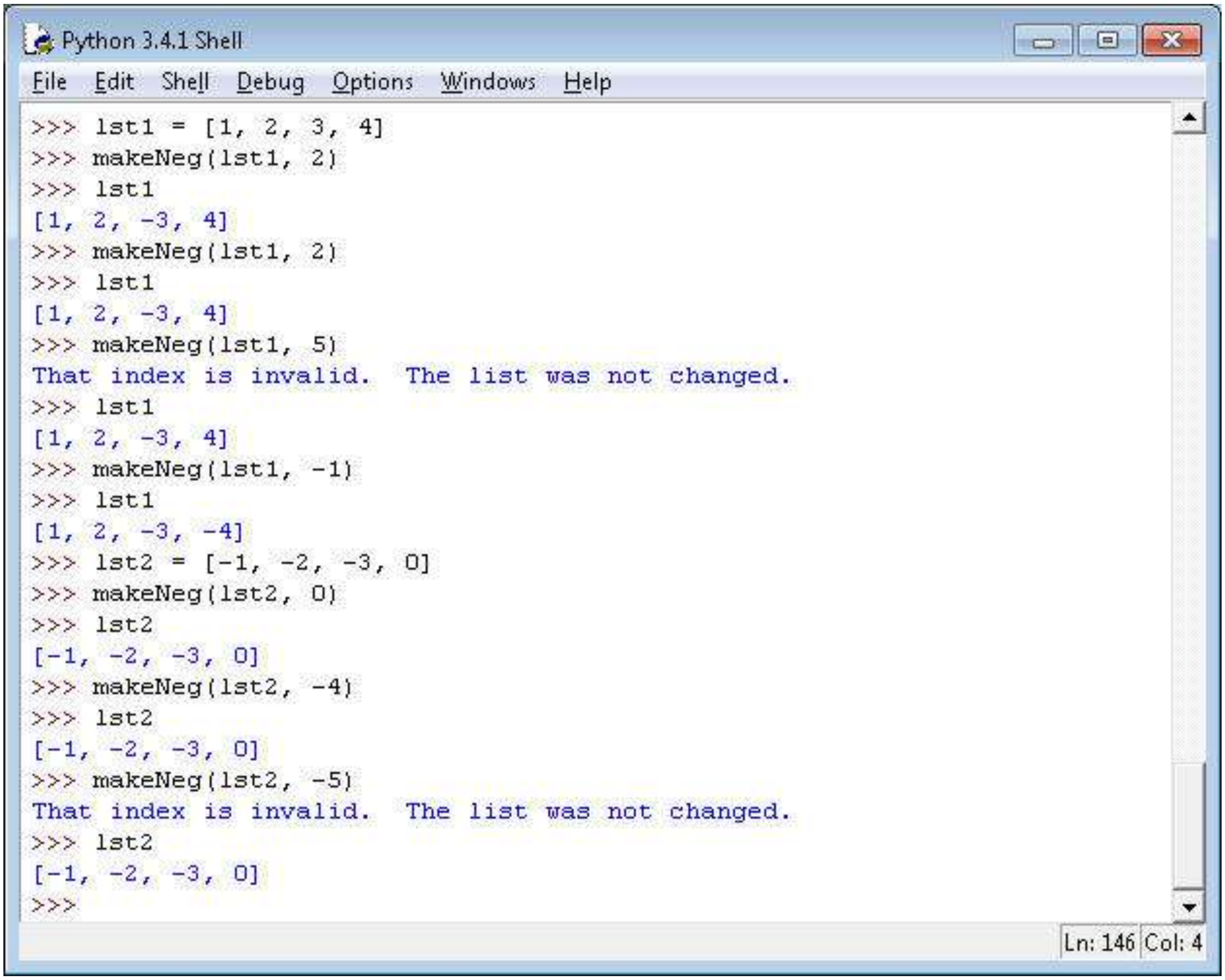
4. Implement a function **numLen**() that takes a string s and an integer n as parameters, and **returns** the number of words in the string s that have length n. Words in the string are non-space characters separated by at least one space. You may assume that the string has

at least one non-space character in it and that the "sentence" does not have punctuation. The following shows several examples of how the function could be used



```
>>> numLen('This is a test', 4)
2
>>> numLen('This is a test', 2)
1
>>> numLen('This is a test', 3)
0
>>> numLen('Zero and the infinite were at the very center', 4)
3
>>> numLen('Zero and the infinite were at the very center', -3)
0
>>> numLen('Zero and the infinite were at the very center', 6)
1
>>> 
```

5. Implement the function **makeNeg**() that takes a list of numbers and a number representing an index as a parameter and changes the list item at the specified index to be negative. If the number at the specified index is already negative, the function shouldn't change the list. The function should verify that the index is correct for the list. If the index isn't valid for the list, the function shouldn't change the list. The information below shows how you would call the function **makeNeg**() and what it would display for several different parameters, including cases where an invalid index is provided:



```
>>> lst1 = [1, 2, 3, 4]
>>> makeNeg(lst1, 2)
>>> lst1
[1, 2, -3, 4]
>>> makeNeg(lst1, 2)
>>> lst1
[1, 2, -3, 4]
>>> makeNeg(lst1, 5)
That index is invalid.  The list was not changed.
>>> lst1
[1, 2, -3, 4]
>>> makeNeg(lst1, -1)
>>> lst1
[1, 2, -3, -4]
>>> lst2 = [-1, -2, -3, 0]
>>> makeNeg(lst2, 0)
>>> lst2
[-1, -2, -3, 0]
>>> makeNeg(lst2, -4)
>>> lst2
[-1, -2, -3, 0]
>>> makeNeg(lst2, -5)
That index is invalid.  The list was not changed.
>>> lst2
[-1, -2, -3, 0]
>>>
```

Submitting the assignment

You must submit the file holding all of the function definitions using the assignment 3 dropbox on the D2L site. Submit only a single Python file (csc241hw3.py) with all of your function definitions inside of it. 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. Each problem is worth 20 points.