4.13 Let string `s` be defined as:

```
s = 'abcdefghijklmnopqrstuvwxyz'
```

Write Python Boolean expressions that correspond to these propositions:

(a) The slice consisting of the second and third character of `s` is `'bc'`.

(b) The slice consisting of the first 14 characters of `s` is `'abcdefghijklmn'`.

(c) The slice of `s` excluding the first 14 characters is `'opqrstuvwxyz'`.

(d) The slice of `s` excluding the first and last characters is `'bcdefghijklmnopqrstuvw'`.

4.14 Translate each part into a Python statement:

(a) Assign to variable `log` the next string, which happens to be a fragment of a log of a request for a text file from a web server:

```
128.0.0.1 - - [12/Feb/2011:10:31:08 -0600] "GET /docs/test.txt HTTP/1.0"
```

(b) Assign to variable `address` the substring of `log` that ends before the first blank space in log, using the string method `split()` and the indexing operator.

(c) Assign to variable `date` the splice of string `log` containing the date `(12/Feb ... -6000)`, using the indexing operator on string `log`.

4.15 For each of the below string values of `s`, write the expression involving `s` and the string methods `split()` that evaluates to list:

```
['10', '20', '30', '40', '50', '60']
```

(a) `s = '10 20 30 40 50 60'`

(b) `s = '10,20,30,40,50,60'`

(c) `s = '10&20&30&40&50&60'`

(d) `s = '10 - 20 - 30 - 40 - 50 - 60'`

4.16 Implement a program that requests three words (strings) from the user. Your program should print Boolean value `True` if the words were entered in dictionary order; otherwise nothing is printed.

```
>>>
Enter first word: bass
Enter second word: salmon
Enter third word: whitefish
True
```

4.20 Given string values for the sender, recipient, and subject of an email, write a string format expression that uses variables `sender`, `recipient`, and `subject` and that prints as shown here:

```
>>> sender = 'tim@abc.com'
>>> recipient = 'tom@xyz.org'
>>> subject = 'Hello!'
>>> print(???)                    # fill in
From: tim@abc.com
To: tom@xyz.org
Subject: Hello!
```

4.23 Write a function `average()` that takes no input but requests that the user enter a sentence. Your function should return the average length of a word in the sentence.

```
>>> average()
Enter a sentence: A sample sentence
5.0
```

4.24 Implement function `cheer()` that takes as input a team name (as a string) and prints a cheer as shown:

```
>>> cheer('Huskies')
How do you spell winner?
I know, I know!
H U S K I E S !
And that's how you spell winner!
Go Huskies!
```

4.25 Write function `vowelCount()` that takes a string as input and counts and prints the number of occurrences of vowels in the string.

```
>>> vowelCount('Le Tour de France')
a, e, i, o, and u appear, respectively, 1, 3, 0, 1, 1 times.
```