Homework #7 - Creating Your Own Module (Library)

Assignment Overview

This assignment focuses on the design and implementation of a library of Python functions to manipulate character strings, as described below.

Assignment Specifications

1. The library module will contain three constants and eight function definitions.

a) It will include the following constants:

```
ASCII_LOWERCASE = "abcdefghijklmnopqrstuvwxyz"
ASCII_UPPERCASE = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
DECIMAL_DIGITS  = "0123456789"
```

b) It will include the following functions:

```
is_alpha ( str ) ==> bool
is_digit ( str ) ==> bool
to_lower ( str ) ==> bool
to_upper ( str ) ==> bool
find_chr ( str, str ) ==> int
find_str ( str, str ) ==> int
replace_chr ( str, str, str ) ==> str
replace_str ( str, str, str ) ==> str
```

The notation above gives the name of each function, the number and type of its argument(s), and the type of its return value.

c) The function names will be spelled exactly as shown (for example, is_digit)

d) Function `is_alpha` has one parameter (a string). It returns **True** if all of the characters in the string are upper case or lower case ASCII letters (it returns **False** otherwise).

e) Function `is_digit` has one parameter (a string). It returns **True** if all of the characters in the string are ASCII decimal digits (it returns **False** otherwise)

f) Function `to_lower` has one parameter (a string). It returns the string which is a copy of the parameter, but where all of the upper case ASCII letters have been converted to lower case ASCII letters.

g) Function `to_upper` has one parameter (a string). It returns the string which is a copy of the parameter, but where all of the lower case ASCII letters have been converted to upper case ASCII letters.

h) Function `find_chr` has two parameters (both strings), where the second parameter must be of length 1. It returns the lowest index where the second parameter is found within the first parameter (it returns -1 if the second parameter is not of length 1 or is not found within the first parameter).

i) Function `find_str` has two parameters (both strings). It returns the lowest index where the second parameter is found within the first parameter (it returns -1 if the second parameter is not found within the first parameter).

j) Function `replace_chr` has three parameters (all strings), where the second and third parameters must be of length 1. It returns the string which is a copy of the first parameter, but where all occurrences of the second parameter have been replaced by the third parameter (it returns the empty string if the second or third parameter are not of length 1).

k) Function `replace_str` has three parameters (all strings). It returns the string which is a copy of the first parameter, but where all occurrences of the second parameter have been replaced by the third parameter. If there are no occurrences of the second parameter in the first, it returns a copy of the first parameter. If the second parameter is the empty string, it returns the string which is a copy of the first parameter, but with the third parameter inserted before the first character, between each character, and after the last character.

l) The library module will not use any of the string methods listed in Section 4.7.1 of the Python Standard Library:
http://docs.python.org/3.3/library/stdtypes.html#string-methods

m) The library module will not contain any **`import`** statements.

n) The library module will not perform any input or output operations. For example, it will not call function `input` or function `print`.

2. Notes

A. ASCII

ASCII abbreviated from **American Standard Code for Information Interchange**, is a character encoding standard for electronic communication. ASCII codes represent text in computers, telecommunications equipment, and other devices. Because of technical limitations

of computer systems at the time it was invented, ASCII has just **128** code points, of which only **95** are printable characters, which severely limited its scope. Modern computer systems have evolved to use **Unicode**, which has millions of code points, but the first 128 of these are the same as the ASCII set.

B. Hint

The following functions from the built-in library may be useful when you implement your library module:

```
ord( str ) == > int # return code of ASCII character
chr( int ) ==> str # return ASCII character of code
```

3. Deliverables

The deliverable for this assignment is the following file:

`library.py` -- your source code solution