COMP 141: String Functions

Assume s, s2, t, and z are string variables, and p and q are integer variables.

len(s)	Returns the length of s.
s + t	Returns the string concatenation of s and t (a new string consisting of all the characters in s, followed immediately by all the characters in t. Note: just doing s + t doesn't change s or t. You must store this new string somewhere if you want to use it later, by saying something like z = s + t (z will be a new string)
s[p]	Returns the character at index p in string s. (Indices start at zero!)
s[p:q]	Returns the substring consisting of all characters in s starting at index p and ending one before index q. Note: if either p or q (or both) is left out, Python will assume p=0 (beginning of the string) and q=len(s) (the end of the string). Ex: s[1:] will return s with the first character left out. Using negative numbers for p and/or q counts from the end of the string: Ex: s[-1] returns the last character in s; s[-2:] returns the last two characters in s.
s in t	Returns True if s occurs as a substring somewhere in t, False otherwise.
s not in t	Returns False if s occurs as a substring somewhere in t, True otherwise.
t.find(s)	Similar to "s in t," but returns the lowest index at which the substring s begins inside of t (assuming it is found somewhere); returns -1 if s is not in t. Think of this as looking through the string t from left to right until s is found (this left to right ordering is important if s occurs more than once in t).
t.find(s, p)	Same as t.find(s), but searches from left to right starting at position p in s.
t.startswith(s)	Similar to "s in t," but only returns True if substring s is found at the beginning of t.
t.endswith(s)	Similar to "s in t," but only returns True if substring s is found at the end of t.
t.replace(s, s2)	Returns a copy of t with all the instances of s replaced by s2.
s.isalnum()	Returns True if s contains only alphanumeric characters and len(s) > 0.
s.isalpha()	Returns True if s contains only alphabetic characters and len(s) > 0.
s.isdigit()	Returns True if s contains only numeric characters and len(s) > 0.
s.islower()	Returns True if all of the alphabetic characters in s are lowercase and len(s) > 0.
s.isupper()	Returns True if all of the alphabetic characters in s are uppercase and len(s) > 0.
s.isspace()	Returns True if all of the characters in s are spaces, tabs, or newlines, and len(s) > 0.
s.lower()	Returns a copy of s with all of the alphabetic characters converted to lowercase. Any non-alphabetic character (or those that are already lowercase) are copied unchanged.
s.upper()	Same as s.lower(), but converts to uppercase.
s.lstrip()	Returns a copy of s with any whitespace characters (spaces, tabs, newlines) removed from the beginning (left side) of the string.
s.lstrip(t)	Returns a copy of s with any characters found in t removed from the beginning of the string.
s.rstrip()	Same as s.lstrip(), but works from the right side (end) of the string.
s.rstrip(t)	Same as s.lstrip(t), but works from the right side (end) of the string.
s.strip()	Combination of s.lstrip() followed by s.rstrip() [removes from beginning & end]
s.strip(t)	Combination of s.lstrip(t) followed by s.rstrip(t) [removes from beginning & end]

Notice that len() is the only function that is not "attached" to a string with a period.