CS2020
Introduction to Programming
Practice Exercise Set 4

String Processing

This exercise set is divided into two parts. In Part 1, you write code fragments to process strings. And, in Part 2, you write functions that accept and/or return strings or a list of strings. You will be ask to implement the code using three techniques: direct access, indexing access with the for loop, and indexing access with the while loop. For each question, it should take less than 15 minutes to derive a solution using one technique.

Do not use map/filter operations or list comprehensions in this set. There is another exercise set for asking you to practice these techniques.

You may use list methods such as find, index, count, and others, unless indicated otherwise.

Practice Topic

- Processing strings using string methods
- Processing strings without using string methods
- Processing strings with functions

Prerequisite

You need to possess knowledge on

- Basic operations such as assignments and expressions
- Input and output (print)
- Data types int, float, and str.
- Simple selection control using if-else
- Basic looping (Practice Exercise Set 1)
- Defining and using functions (Practice Exercise Set 2)

Example 1

S is a string. Count how many characters in S are uppercase alphabets A to Z.

a. Direct Access Loop

```
cnt = 0
for ch in S:
    if ch.isupper(): ## if ch in "ABCDEFGHIJKLMNOPQRSTUVWXYZ":
        cnt += 1
```

b. Indexing Access with for

```
cnt = 0
for idx in range(len(S)):
   if S[idx].isupper(): ## if ch in "ABCDEFGHIJKLMNOPQRSTUVWXYZ":
      cnt += 1
```

c. Indexing Access with while

```
cnt, idx = 0, 0
while idx < len(S):
   if S[idx].isupper(): ## if ch in "ABCDEFGHIJKLMNOPQRSTUVWXYZ":
      cnt += 1
   idx += 1
```

Example 2

Write a function count_vowels that accepts one string parameter S. Return the number of the vowels (both upper and lowercase) found in S.

a. Direct Access Loop

```
def count_vowels(S):
   cnt = 0
   for ch in S:
      if ch in "AEIOUaeiou": cnt += 1
    # if ch.upper() in "AEIOU": cnt += 1
   return cnt
```

b. Indexing Access with for

```
def count_vowels(S):
   cnt = 0
   for idx in range(len(S)):
      if S[idx] in "AEIOUaeiou": cnt += 1
    # if S[idx].upper() in "AEIOU": cnt += 1
   return cnt
```

c. Indexing Access with while

```
def count_vowels(S):
   cnt, idx = 0, 0
   while idx < len(S):
      if S[idx] in "AEIOUaeiou": cnt += 1
    # if S[idx].upper() in "AEIOU": cnt += 1
      idx += 1
   return total
```

Part 1 Code Fragments

For each question, provide a solution using direct access, indexing with the for loop, and indexing with the while loop, unless indicated otherwise.

1. Create a list of strings by inputting strings from the user. Use the while loop to repeatedly prompt the user for input and stop the repetition when the user enters the string stop. The stop keyword is case sensitive; all letters must be lowercase.
2. Create one long string by concatenating all strings entered by the user. Put one blank space between the strings entered by the user. Use the while loop to repeatedly prompt the user for input and stop the repetition when the user enters the string stop. The stop keyword is case sensitive; all letters must be lowercase. Do not include stop in the final result.
3. Create a list of strings that have 5 or more vowels, both upper and lowercase. Strings entered the user. Use the while loop to repeatedly prompt the user for input and stop the repetition when the user enters the string stop. For this exercise, the stop keyword is case insensitive; the letters can be either upper or lowercase.
4. Given a string S, replace all lowercase alphabets to 'X'. Do not use the replace method here.
5. Given a string S, replace all lowercase alphabets to uppercase and all uppercase alphabets to lowercase. A becomes a, a becomes A, B becomes b, and so forth. Do not use the replace method here.
6. Given a string S that contains only alphanumeric characters (i.e., no punctuations or symbols), extract the substrings that represent integers and compute their sum. For example, if S is "I have 100 pens and 34 apples" then you extract "100" and "34" and compute their numerical sum 134.
7. Given an integer N, convert it to a string representation with commas. For example, if N is 14500986, then convert it to "14,500,986".
8. Given a string S that represent a numerical value with commas, convert it to an int. For example if S is "14,500,986", convert it to an int 14500986.
9. Print out a given string message N times. The first line has zero blank space before message, the second line 5 blank spaces before, the third line 10 blank spaces, and so on. The N'th line has 5*(N-1) blank spaces before the message text. Output looks something like this

```
Hello World
     Hello World
          Hello World
```

10. Given a string S, print out one character from both ends per line. When S is "HELLO" the output will be

```
H   o
e   l
```

```
l  l
l  e
o  H
```

Part 2 Functions

For each question, provide three versions of a function definition with a) direct access looping, b) indexing access with for, and c) indexing access with while.

1. Write a function greeting that returns a message Hello, <name>. How are you? where <name> is a string value passed to the function.
2. Write a function get_vowel_cnt that returns that returns the number of vowels, both upper and lowercase, found in the parameter word.
3. Write a boolean function has_operator that accepts a string S and returns True if S includes an arithmetic operator +, -, *, and /.
4. Write a function get_words that accepts one string sentence and returns a list of words (strings) in sentence. If sentence is "I am doing great" then the function returns ["I", "am", "doing", "good"]. Assume there are no punctuation marks or symbols. Do not use any string methods here.
5. Write a function get_words that accepts one string sentence and returns a list of words (strings) in sentence. If sentence is "I am doing great" then the function returns ["I", "am", "doing", "good"]. Assume there are no punctuation marks or symbols. This time, you may use any string methods.
6. Write a function replace_lower that accepts a string S and returns a new string with all lowercase alphabets in S replaced by 'X'. Do not use the replace method here.
7. Write a function switch_lower_upper that accepts a string S and returns a new string by replacing all lowercase alphabets in S to uppercase and all uppercase alphabets in S to lowercase. A becomes a, a becomes A, B becomes b, and so forth. Do not use the replace method here.
8. Write a function compute_sum that accepts a string S that contains only alphanumeric characters (i.e., no punctuations or symbols) and returns the sum of numbers found in S. For example, if S is "I have 100 pens and 34 apples" then you extract "100" and "34" and return their numerical sum 134.
9. Write a function get_comma_format that accepts an integer N and returns a string representation of N with commas. For example, if N is 14500986, then return the string "14,500,986".
10. Write a function convert_to_int that accepts a string S, a string representation of a numerical value with commas, and returns the corresponding int value. For example if S is "14,500,986", return an int 14500986.