

**Task1** Create a function named **is\_nth\_char** that takes 3 parameters:

- char: a character
- n: an integer representing a string index
- str: a string

The function should return a boolean value indicating whether the string contains the character at the specified index. The function should be case sensitive (i.e. 'b' is not a character in 'Birthday') and should not cause index out of bounds errors. Run the unit tests to check your function.

**Task2** Create a function named **count\_after** that takes 3 parameters:

- search\_char: a character
- start\_char: a character
- str: a string

The function should return a count of the number of times the search character exists in the substring that begins after the first occurrence of the start character. So if the start character is 'p' and the string is 'Happy birthday', the substring to count occurrences of the search character would be 'ppy birthday'. The function should be case sensitive and should not cause index out of bounds errors. Run the unit tests to check your function.

**Task3** Create a function named **cutout** that takes 3 parameters

- str: a string
- start: a string index
- end: a string index

The function should return the concatenation of the substring before the start index with the substring after the end index. The function should not cause index out of bounds errors. Return the empty string if the start or end indices are out of bounds, or if the start index exceeds the end index. Run the unit tests to check your function.

**Task4** Create a function named **upper\_substring** that takes 3 parameters

- str: a string
- start: a string index
- end: a string index

The function should return a modified version of the string passed as parameter, with the substring indexed by start and end (inclusive) converted to upper case letters. The function should not cause index out of bounds errors. Return the empty string if the start or end indices are out of bounds, or if the start index exceeds the end index. Run the unit tests to check your function.

**Task5** Create a function named **count\_digits** that takes 1 parameter

- str: a string

The function should return a count of the number of digits contained within the string. Run the unit tests to check your function.

**Task6** Create a function named **sum\_digits** that takes 1 parameter

- str: a string

The function should return the sum of the digits contained within the string. (HINT: Convert each character that is a digit to an integer). Run the unit tests to check your function.

**Task7** Create a function named **convert\_name** that takes 1 parameter:

- str: a string

Many documents use a specific format for a person's name.

(1) If the parameter string is of the form: 'first middle last'

the function should return a string with the last name followed by first and middle initials: 'last, f.m.'

(2) If the parameter string is of the form: 'first last'

the function should return a string with the last name followed by first initial: 'last, f.'

(3) If the parameter string does not contain a space, the function should return the original parameter string.

Run the unit tests to check your function.

## **SUBMIT TO CANVAS**

Submit hw9.ipynb. Save your notebook. Select File/Download As/Notebook.

Submit hw9.pdf. Select File/Print Preview. Right click/Print/Save as pdf.