"Introduction to Programming" - Lab 7

Exercise Sheet – Practice Session (our last practice session)

This exercise sheet examines the concepts covered in recent lectures (as well as content we may cover this week).

Setup:

1. Note: anywhere I write <text> (text enclosed in a pair of angle brackets), I want you to change this text to the text I'm looking for (making sure to remove the angle brackets), so for example when I say:
    a. # <todays date>, you write:
    b. # Tuesday 12-10-2021
2. Create a new Folder called "Programming_Folder", if it does not exist. This will be your course Project folder.
3. Download from Canvas the zip file " Lab<number>.zip", unzip and place the contents in the "Programming_Folder" folder.
4. Rename Lab<number> to the number of this Lab
5. The unzipped folder contains, two Python files:
    a. my_functions.py
        i. my_functions.py - is where you will place all the functions created for this lab
    b. main.py
        i. main.py - is the primary file you will call when testing your code
        ii. all of the functions created in 'my_functions.py', will be called/invoked from your 'main.py' file
    c. In these files, I give one example where I create a function called 'print_function()' in my_functions.py and I call it from main.py
6. You should understand all of the code in these two files, if not ask me and I will explain
7. Make sure you add your name and student number to both 'main.py' and my_functions.py'
8. Once complete, and prior to the submission deadline, please upload your updated version of 'functions.py' to Canvas. Do not upload your version of 'main.py', as I will test the code in your 'my_functions.py' file, with my own code.

Exercises:

1. Write your own version of the following Python functions, using while loops when appropriate, and add comments to your code (docStrings and comments per line of code). Do not use the respective original python function calls. Example of what is returned by the function call is shown:
    a. count(my_list, value) - function to return the number of times *value* occurs in *my_list*
        i. count("hello", "l") -> 2 (the integer 2 is returned)

    b. index(my_list, value) - function to return the first index that *value* occurs in *my_list*. Return -1 if the *value* does not occur in *my_list*
        i. index("hello", "o") -> 4
        ii. index("hello", "p") -> -1

    c. get_value(my_list, index) - function to return the value that occurs in *my_list* at *index.* If index does not return a valid value from my_list, it should return the error message as outlined in item 3 below.
        i. get_value("hello", 1) -> "e"

    d. insert(my_list, index, value) - function to return my_list, after you have added value at index (remember to check the length of my_list and if index is larger than len(my_list) add as a new index at the end my_list)
        i. insert("hello", 1, "a") -> "hallo"
        ii. insert("hello", 5, "p") -> "hellop"

    e. value_in_list(my_list, value) - function to return True or False if value occurs in my_list
        i. we can then use "if value_in_list(list, value):" as a boolean check
        ii. value_in_list("hello", "o") - True
        iii. value_in_list("hello", "a") - False

    f. concat(list1, list2) - function to return a new list, which is a combination of list1 and list2
        i. concat("hello", " world") -> "hello world"

    g. remove(value, my_list) - function to return a list with the first occurrence of value removed from my_list
        i. remove("h", "hello") -> "ello"

2. To give you an example of how the code could be written I have provided the code and some calls for *count(my_list, value).* I have also commented the code. No Error handling is added to my version of the function

3. For all of these functions, if an error occurs return the error message “Houston, we have a problem!”. No Error handling is added to my version of the count function. Make sure you add error handling to your submission of count() and all other functions in this assignment.

Error Checking:

1. There is no error checking in this lab, but...
   a. I want you to try and break your code :)
   b. Change the inputs to the wrong type, the wrong value, and document what happens as a comment in your code.
   c. Try the break challenge for each function you have created. For now you only need to try to break each function once, and then fix for this once.