CIT 590 - HW7

Submit this along with HW6.

This HW is different from any of the others this semester.
Your instructions are

- You have to do it individually. Do not plagiarize! Although it is due with HW6 which is potentially a pair programmed HW, we want you to submit this individually.
- You do not need to have unit tests.
- You do not have to check for incorrect input. Assume you will get the type of arguments that you expect.
- You do not have to write any kind of main function.
- For questions 4, 5 and 6 - the emphasis is on writing the minimal number of lines. The fewer lines you write, the better.

1. Write a recursive function that checks to see if an input string is of the form $aaaa \ldots abbbbb \ldots b$. The block of as has to be same length as the block of bs.

   Let the function be called **sameAB.**

   sameAB('aaabb') returns False

   sameAB('aaabbb') returns True

   sameAB('') returns True

   sameAB('a') returns False

2. You are given a sorted list of integers. Write a function that recursively implements binary search.

   binary search works like this. Assume the list is in ascending order. In each step, the algorithm compares the search value with the value of the middle element of the list. If the values match, then a matching element has been found and its index, or position, is returned. Otherwise, if the value being searched for is less than the middle element, then the algorithm repeats its action on the sub-list to the left of the middle element or, if the search key is greater, on the sub-list to the right.

   This function just needs to return True or False depending upon whether the value being searched for is in the list or not.

   binary_search(lst, val)

3. Recursively flatten a list which contains a collection of lists.

   lst = [[1,2,3], [4,5], [6,7,8]]

   flatten(lst) = [1,2,3,4,5,6,7,8]

   Write this **flatten** function. It should take a list of lists and spit out a single list.

4. Write a single line of code that

   Creates a list of initials of strings being passed in as names

   So a list of ['Mike Tyson', 'Marilyn Monroe', 'George Bush'] should get converted to ['M.T.', 'M.M', 'G.B.']

   Assume you have a variable called lst which contains a list of names.

   Write a function called **initials(lst)**

5. You are given a csv file that contains the list of students in the class. Assume that you have the following columns in the csv file - Name, Department, Email.

   You want to find the number of students who are from MEAM.

   Do this using list comprehension in the minimum number of lines.

   Write a function called **meamers(filename)** where filename is the name of a csv file which has this information. You just have to return the number of MEAM students

6. You are given a dictionary which has the alphabets as keys and the number of occurrences of each alphabet as the values.

   For example

   freq = {'a' : 3, 'b' : 4, 'c' : 5, 'd' : 2, 'e':13, 'f': 2}

   Using the minimum lines of code (hint: use list comprehension), find the alphabet that is appearing the most times.

   Write a function called **most_frequent_alphabet(frequency_dictionary)**

What to submit

Submit one single file called HW7.py with the following 6 functions

sameAB(abstring)

flatten(lst_of_lst)

binary_search(lst, val) - this returns true or false depending upon whether the value is in the list lst. Assume integer list.

most_frequent_alphabet(frequency_dictionary)

meamers(filename)

initials(lst)