We have not provided tests for any of the following questions. You may test them with any of your own input or any example input we did provide. If you would like to make your own tests for practice (hint hint wink wink nudge nudge) feel free.

1. Write a function that accepts two parameters, the first being filename string and the second being a string that contains just a single character. The file will be a csv that looks like

Name, Favorite Color, Age, Address

first last, color, age, addr1 \n addr2

first last, color, age, addr1 \n addr2

…  
  
Your function should open and parse the file. Read every line in the file and return an array containing the name of every person whose last name contains the specified character (not case sensitive).

Assume you have access to your array util functions

1. Using that same file create a function that accepts function parameters of a filename string and a color as a string. This function should return every state that has an inhabitant with the specified color as their favorite. For example if the parameter “Orange” is passed in then return a data structure containing CT, CA, MA.

You can pick the data structure you use! but you should probably pick Array, List, or Tuple

Don't have any duplicate states

1. Factorials are a function in math that multiplies a number by every number smaller than it that's greater than 0.. For example:

5! = 5 \* 4 \* 3 \* 2 \* 1

6! = (10 \* 5 \* 4 )+ (10 \* 2 \* 1)

Write a recursive function that accepts a number as a parameter and returns the factorial of that number.

1. Create a recursive function that accepts a string and returns True or False on whether or not that string is a palindrome. A palindrome is a word that is the same forwards and reversed.

Racecar -> True

palindrome -> False

Never Odd Or Even - > True

1. You may have seen that in a python list, you can actually store another list. Doing this at one level creates a 2d list, however there is no reason that list can’t contain even more lists. This can go on for a while and you can have a very large structure of lists.

Given a Python list that can contain another list, write a recursive function that can add every number in all of the lists no matter the depth.

Use len(item) to determine whether the item you're checking is another list or an integer

len(int) will give a TypeError that you can handle

Note: Its generally not good practice to use handling errors as part of your logic, but were just doing this for practice with handling errors

[1, 2, [3, 4, [5, 6], 7], 8] -> 36

1. Write a function that takes a number n and returns a list of square roots of every number 1 to n

b) Write a function that does the same thing but uses list comprehension.

1. Create a function that takes two lists as parameters. Return one list with each of the items in the second appended onto each of the items in the first.

[ “help “, “smell “]

[ “yourself”, “me”] -> [“help yourself”, “help me”, “smell yourself”, “smell me”]

1. Write a function that accepts a string and will return True or False based on if the provided string is a standard rit student email address. Assume that the string will have the form of XXX-@YYY-.ZZZ where a “-” means any number of characters

There could be any number of characters before the @

You need to check for 3 letters, then 4 numbers, then EITHER @rit.edu OR @g.rit.edu

[sec7185@rit.edu](mailto:sec7185@rit.edu) -> True

[sec7185@g.rit.edu](mailto:sec7185@g.rit.edu) -> True

[stephen@rit.edu](mailto:stephen@rit.edu) -> False

Hint: Use Regex (Regular expressions) and you can find helpful regex testers online