**Unit Two Lab**

Directions: Complete each of the programs below and submit them by the date specified in the assignment sheet and Blackboard. Submit **copies of your code** and **screenshots of the code running with each task**. Use your notes and scripts from your Python class and the website, <https://w3schools.com/python> or other Internet references as reference guides for this review assignment. Also, be sure to use the document, **Script Requirements as a guide** to writing good code. **Full credit will not be earned if you do not meet these script requirements.** The one exception is that in the first two weeks, your code does not need to completely validate all inputs unless specified. **40 points**

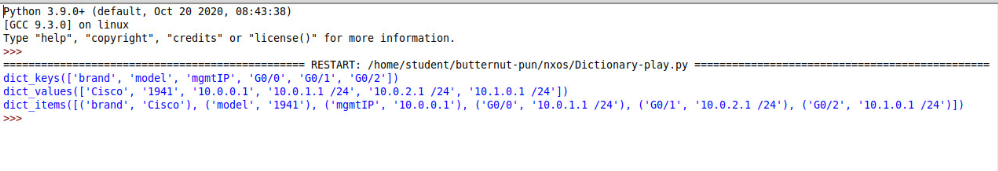
1. Python dictionaries are one good way to organize data. Review dictionaries at the above website before continuing. Consider the dictionary below:

router1 = {  
  "brand": "Cisco",  
  "model": "1941",  
  "mgmtIP": “10.0.0.1”  
}

* 1. Note the format: Curly braces and key: value pairs, with pairs separated by commas. Note that key names and values are in quotes.
  2. To access a value, you refer to a key:
     1. ip = router1[“mgmtIPAddress”]
     2. The above assigns 10.0.0.1 to the variable, ip
  3. Using a dictionary, create one for router1 that contains the above information as well as the following key: value pairs:
     1. Three additional interfaces **as keys**, G0/0, G0/1, and G0/2 each with the following respective information stored for their **values,** 10.0.1.1 /24, 10.0.2.1 /24, and 10.0.3.1 /24.
     2. A key: value pair of “hostname”: “r1”
  4. Modify your program, using the above dictionary, that displays your keys, values, and items of your dictionary (e.g., print(router1.keys(), etc.). See Dictionary Methods in the website listed in the directions or another resource for assistance. Be sure to send a screenshot of the outputs. Note the various outputs and their formats. Your output should appear like the following (**note some key: value pairs have been modified for the next section and your output values will differ slightly**):

Graphical user interface, text, application

Description automatically generated



* 1. Modify the above program, and this time, change the G0/2 IP address in the script to 10.1.3.1 /24 before it prints out. For example, if you were to change the brand value to HP, you would type: *router1[“brand”] = “HP”* Do not re-create the dictionary object, but instead modify the value for the key pair from 10.0.3.1 to the address specified. Also, change the model from 1941 to 2901. Refer to the Access Items and Change Items sections in the Website listed above or another reference for how to change keys and values.
  2. Re-run the script. Be sure to send the code and a screenshot of the output.

Graphical user interface, text, application

Description automatically generated

1. Using the dictionary object above, write a program that loops through the dictionary items (i.e., router1.items()) with a For Loop, and displays an output similar to the one below:

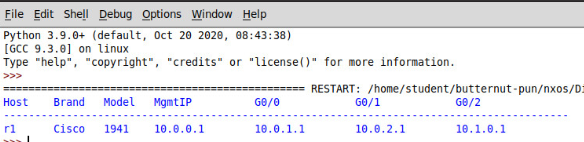
Graphical user interface, text, application, email

Description automatically generated

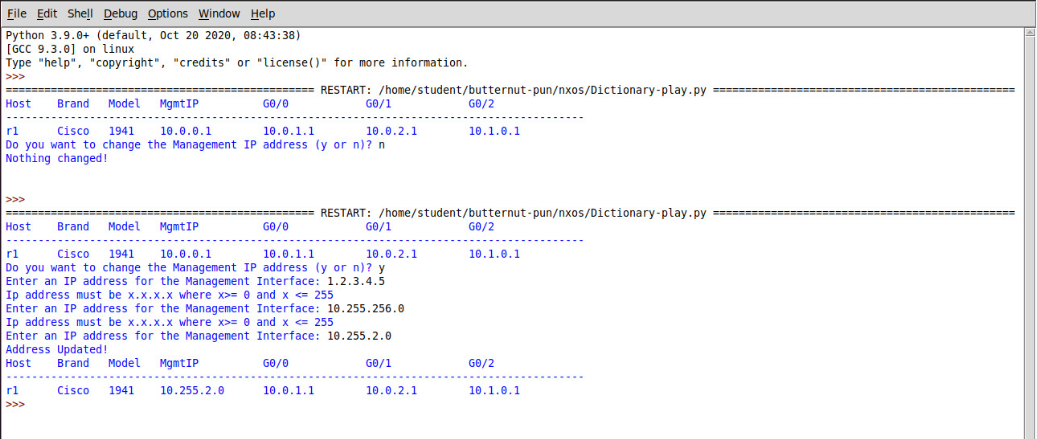
**Hint: See Loop Dictionaries section on the Website in the directions above for reference**

1. Using the dictionary object above, write a program that displays the following in a similar format. Note that the interfaces g0/0-2 do not have the CIDR displayed, even though it is stored in the dictionary. You will need to remove that from your string, using string methods like split() or replace()

Hint: You can use replace() to simply remove “/24” and replace it with an empty string” “ or in a more sophisticated manner, use split() to display only the first element of the returned list, which allows you to remove any CIDR notation.

**Hint: You can concatenate a ‘\t’ (in some cases two tabs are needed) to concatenate a tab to line up your fields**

1. Using the dictionary object above, write a program that:
   1. Displays the dictionary object as in step 3.
   2. Then have the script ask the user if they want to change the Mgmt IP address. If the user responds in the negative, then end the script. If the user responds in the affirmative, then ask them for the new IP address. Verify that the new value is a valid IP address. If it is not, give the user a message telling them the requirements for a valid IP address (e.g., x.x.x.x where x is a number from 0-255). If the address is valid, change the address value in the dictionary and re-display the output as in step 3, above. Below is a sample of the running program:



**Hint: Use a while loop when determining a valid IP address. Prompt the user for an address in the loop and have the loop end when a valid IP is entered. Test for the IP within the loop. Use input to store the candidate IP address as a string. Use the split() method for the string, but instead of using the default space as a separator, use the period as in string.split(“.”). Remember this returns a list. You can use the len() to see how many items are on the list (octets) to ensure there are exactly 4 and you can then iterate the list, checking each element for being in a range from 0 -255. If you are feeling lucky, you can use a regular expression instead. Don’t forget to convert the iterated value to an integer before checking its value. Also, you can earn a bonus 10 points if you check the validity of the IP address in a function.**

1. **Organize all of your answers and screenshots, placing them with the questions above and submit to Blackboard by the due date. To receive full credit, you must complete the entire lab.**