Highlight Duplicate Device Information

In this exercise, you will read the information from the devices file in the PRNE/section10 folder and highlight any duplicate IP addresses found. In order to write this application, you will:

- Use a `for` loop to read in device information
- Use a set to store known IP addresses
- Do a comparison of each device IP address against a set of known IP addresses.
- Use a continue statement to move to the next iteration if a match is found.

Step 1

Read device information from the file and store in a list.

Answer

```
devices_list = [] # Create the outer list for all devices

# Read in the devices from the file
file = open('devices','r')
for line in file:

    device_info_list = line.split(',') # Get device info into list
    devices_list.append(device_info_list)

file.close() # Close the file since we are done with it
```

Step 2

Print the device information in a table. If an IP address is a duplicate of one seen before, print a message indicating it is a duplicate.

Your output should look similar to:

```
Name     OS-type    IP address           Software
-------  ---------  -------------------  --------------------
d01-is   ios        Mgmt:10.3.21.5       Version 5.3.1
d02-is   ios        Mgmt:10.3.21.6       Version 4.22.18
d03-nx   nx-os      Mgmt:10.3.21.7       Version 5.3.1
d04-nx   nx-os      Mgmt:10.3.21.8       Version 5.3.1
d05-xr   ios-xr     Mgmt:10.3.21.8       Version 4.16.9     *Duplicate IP*
d06-xr   ios-xr     Mgmt:10.3.21.10      Version 5.3.0
d07-xe   ios-xe     Mgmt:10.3.21.19      Version 4.16.0
d08-xe   ios-xe     Mgmt:10.3.21.22      Version 5.3.0
```

Answer

```
print ''
print 'Name     OS-type    IP address           Software            '
print '-------  ---------  -------------------  --------------------'

ip_addresses = set()

# Go through the list of devices, printing out values in nice format
for device in devices_list:

    print '{0:8} {1:10} {2:20} {3:20}'.format(device[0],device[1],device[2],

    # Print 'duplicate' if IP address exists for another device
    if device[2] in ip_addresses:
        print '*Duplicate IP!*'
        continue

    ip_addresses.add(device[2])
    print ''

print ''
```

Select Devices by IP Address

In this exercise, you will write an application that allows a user to search for device information by entering the IP address. If the IP address is found, the application will display the device information. The application will iterate until the user enters Ctrl-C. In order to write this application, you will:

- Use a `for` loop to read in device information
- Use a `while` loop to allow users to input IP addresses to search for
- Use `range` to iterate through devices in the list
- Use `continue` to continue searching the list
- Use `break` to exit when user presses Ctrl-C
- Use `for...else` to print if the list was exhausted without finding a match

Here are some Python tools that you will find helpful to complete this lab:

- **Input:** Because the Python 2.7, `input` function expects a Python expression as input, and IP addresses are not Python expression, use `raw_input`:

```
ip_address = raw_input('Enter device IP address to find (Ctrl-C to exit):')
```

- To capture the Ctrl-C, surround your input code with a `try-except` routine, catching the exception called `KeyboardInterrupt`:

```
try:
    ip_address = raw_input('Enter device IP address to find (Ctrl-C to exit)
except KeyboardInterrupt:
     break;
```

- Recall that the IP address format for these current labs follows the format 'Mgmt:<ip-address>'. To specify a subset of a string, specify the start and end locations. The following statement is from the solution to this lab, specifying that you are looking at `device_info[2]`, which is the second item, or the IP address. The `[5:]` specifies that you are looking at the substring starting at the fifth character, until the end.

```
if device_info[2][5:] == ip_address:  # Check to see if device IP is a match
```

Step 3

Read the devices file, creating a list of devices, with each device holding a list of device information. Print a table of the information as you read it in.

Answer

```
devices_list = [] # Create the outer list for all devices

print ''
print 'Idx Name     OS-type  IP address           Software            '
print '--- -------- -------- -------------------- --------------------'

index = 0

# Read in the devices from the file
file = open('devices','r')
for line in file:

    device_info = line.split(',') # Get device info into list
    devices_list.append(device_info)

    print '{0:2}: {1:8} {2:8} {3:20} {4:20}'.format(index+1,
                                                   device_info[0],device_in
                                                   device_info[2],device_in


    index += 1 # increment our index

file.close() # Close the file since we are done with it
print ''
```

Step 4

Loop forever (while true) reading user input, and matching their input IP address with an item from the list of devices. If the user enters Ctrl-C, exit the program. If the user enters anything else, use it to search for a matching IP address from the list of devices. Use a for statement with a range. The range will go from 0 to the length of the list. If the IP address is found, print a message with the device data. If the IP address is not found, print a message stating: "--- Given IP address not found ---".

Answer

```
while True: # Loop forever, until user terminates program

    # Request user to input the IP address we will search for
    try:
        ip_address = raw_input('Enter device IP address to find (Ctrl-C to e
    except KeyboardInterrupt:
        break;

    # Loop through our devices looking for a match on IP address
    for index in range(0, len(devices_list)):

        device_info = devices_list[index] # Get information for this device
        if device_info[2][5:] == ip_address:  # Check to see if device IP is

            # If a match, print results and stop looking
            print '{0:2}: {1:8} {2:8} {3:20} {4:20}'.format(index+1,
                                                    device_info[0],device_i
                                                    device_info[2],device_i
            break
        else:
            continue

    else:  # We get here if we exhausted the device list, IP not found
        print '--- Given IP address not found ---'

print '\n'
print 'Device search terminated.\n'
```