Creating Functions That Return Values

In these exercises, you will create functions that return values.

Create a Connect Function

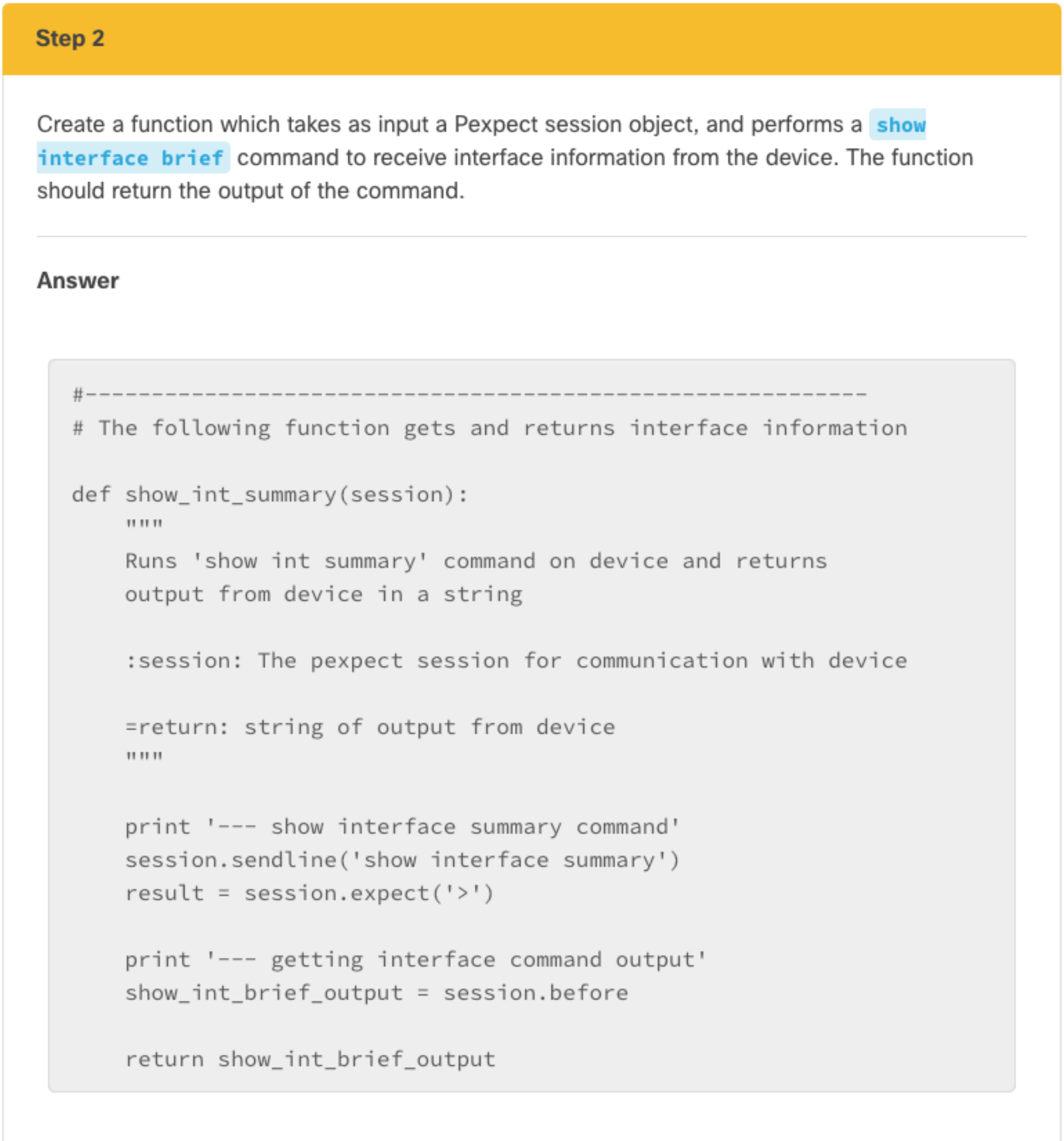
In this exercise you will:

- Create a function to connect to a device in your virtual network, returning the Pexpect 'session' object.
- Create another function to use the session object to read interface information from the device, returning the output of the `show int brief` command.
- Your main code will call the connect function, call the show interface function, and print the output just as it was received from the device.

Step 1

Create a function which takes the IP address, username, and password of the device as input parameters. The function should use Pexpect to connect to the device. If successful, the function should return the Pexpect 'session' object. If unsuccessful, the function should return 0.

```
import pexpect

#-----------------------------------------------------------
# The following code connects to a device

def connect(dev_ip,username,password):
    """
    Connects to device using pexpect

    :dev_ip: The IP address of the device we are connectin to
    :username: The username that we should use when logging in
    :password: The password that we should use when logging in

    =return: pexpect session object if succssful, 0 otherwise

    """

    print '--- attempting to: telnet ' + dev_ip

    session = pexpect.spawn('telnet ' + dev_ip, timeout=20)

    result = session.expect(['Username:', pexpect.TIMEOUT])
    # Check for failure
    if result != 0:
        print '--- Timeout or unexpected reply from device'
        return 0

    print '--- attempting to: username: ' + username

    # Successfully got username prompt, logging with username
    session.sendline(username)
    result = session.expect(['Password:', pexpect.TIMEOUT])
    # Check for failure
    if result != 0:
        print '--- Timeout or unexpected reply from device'
        return 0

    print '--- attempting to: password: ' + password

    # Successfully got password prompt, logging in with password
    session.sendline(password)
    session.expect('>')

    return session  # return pexpect session object to caller
```

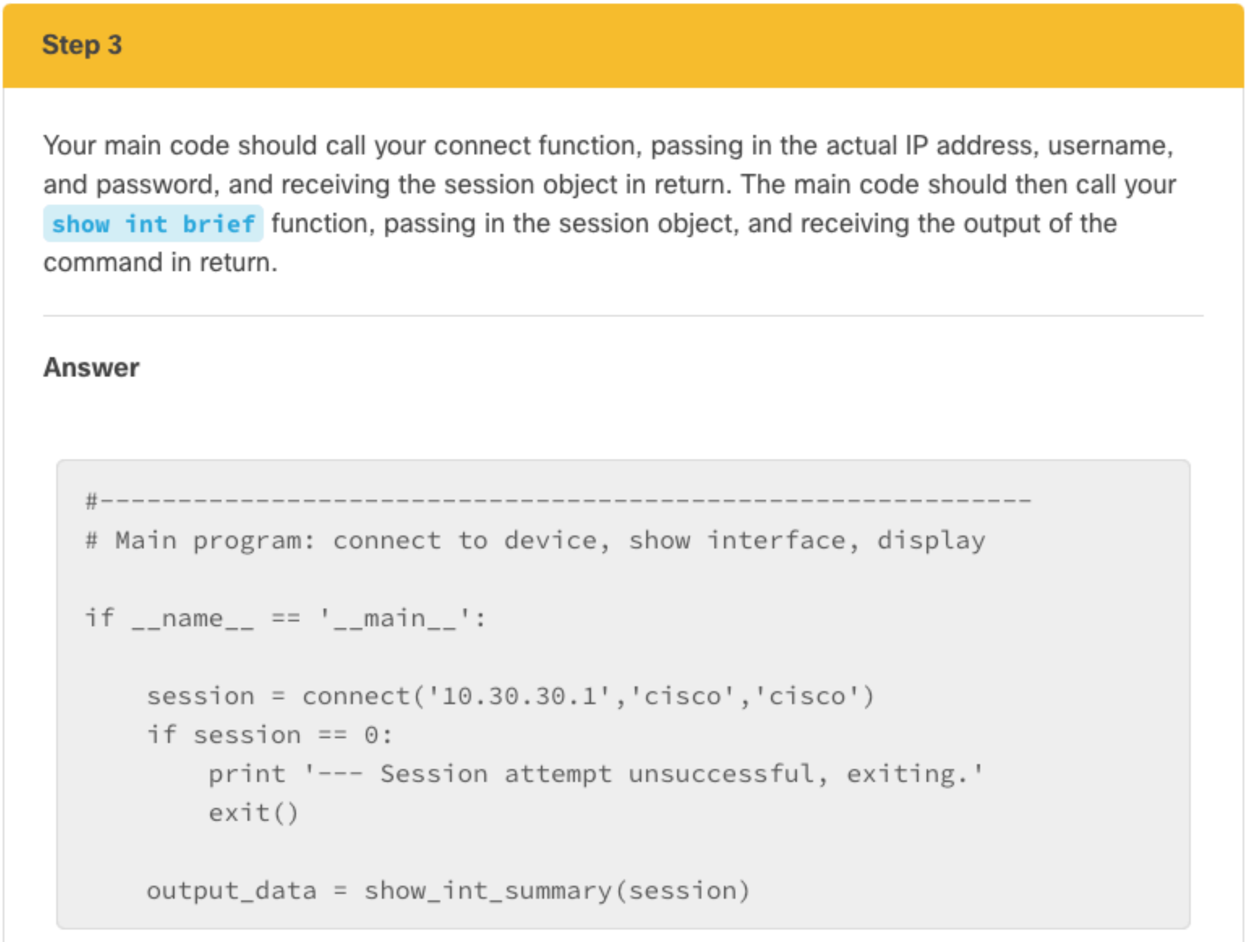


Step 2

Create a function which takes as input a Pexpect session object, and performs a `show interface brief` command to receive interface information from the device. The function should return the output of the command.

Answer

```
#-----------------------------------------------------------
# The following function gets and returns interface information

def show_int_summary(session):
    """
    Runs 'show int summary' command on device and returns
    output from device in a string

    :session: The pexpect session for communication with device

    =return: string of output from device
    """

    print '--- show interface summary command'
    session.sendline('show interface summary')
    result = session.expect('>')

    print '--- getting interface command output'
    show_int_brief_output = session.before

    return show_int_brief_output
```

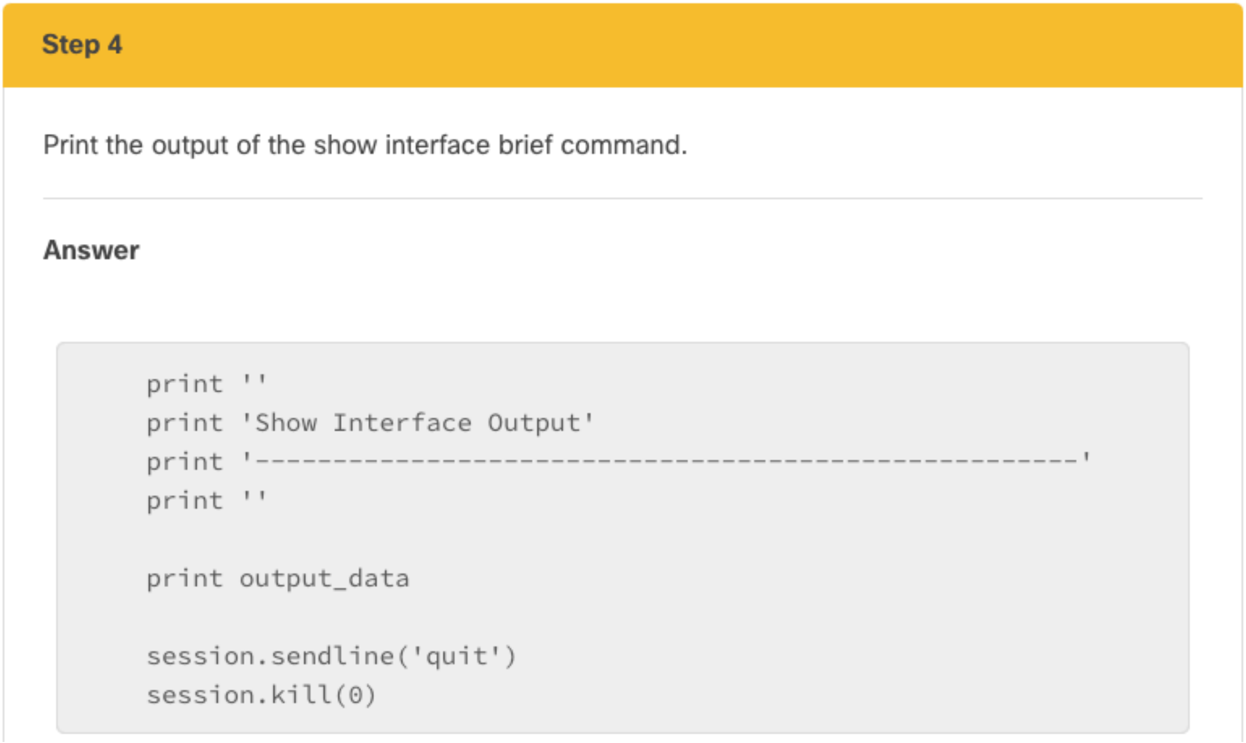


Step 3

Your main code should call your connect function, passing in the actual IP address, username, and password, and receiving the session object in return. The main code should then call your `show int brief` function, passing in the session object, and receiving the output of the command in return.

Answer

```
#------------------------------------------------------------
# Main program: connect to device, show interface, display

if __name__ == '__main__':

    session = connect('10.30.30.1','cisco','cisco')
    if session == 0:
        print '--- Session attempt unsuccessful, exiting.'
        exit()

    output_data = show_int_summary(session)
```



Step 4

Print the output of the show interface brief command.

Answer

```
    print ''
    print 'Show Interface Output'
    print '-----------------------------------------------------'
    print ''

    print output_data

    session.sendline('quit')
    session.kill(0)
```

Create an Inventory Function

Step 5

Create a function that reads the device information from the file real-devices located in the PRNE/section11 folder. Return a list of devices, with information for each device stored in a dictionary.

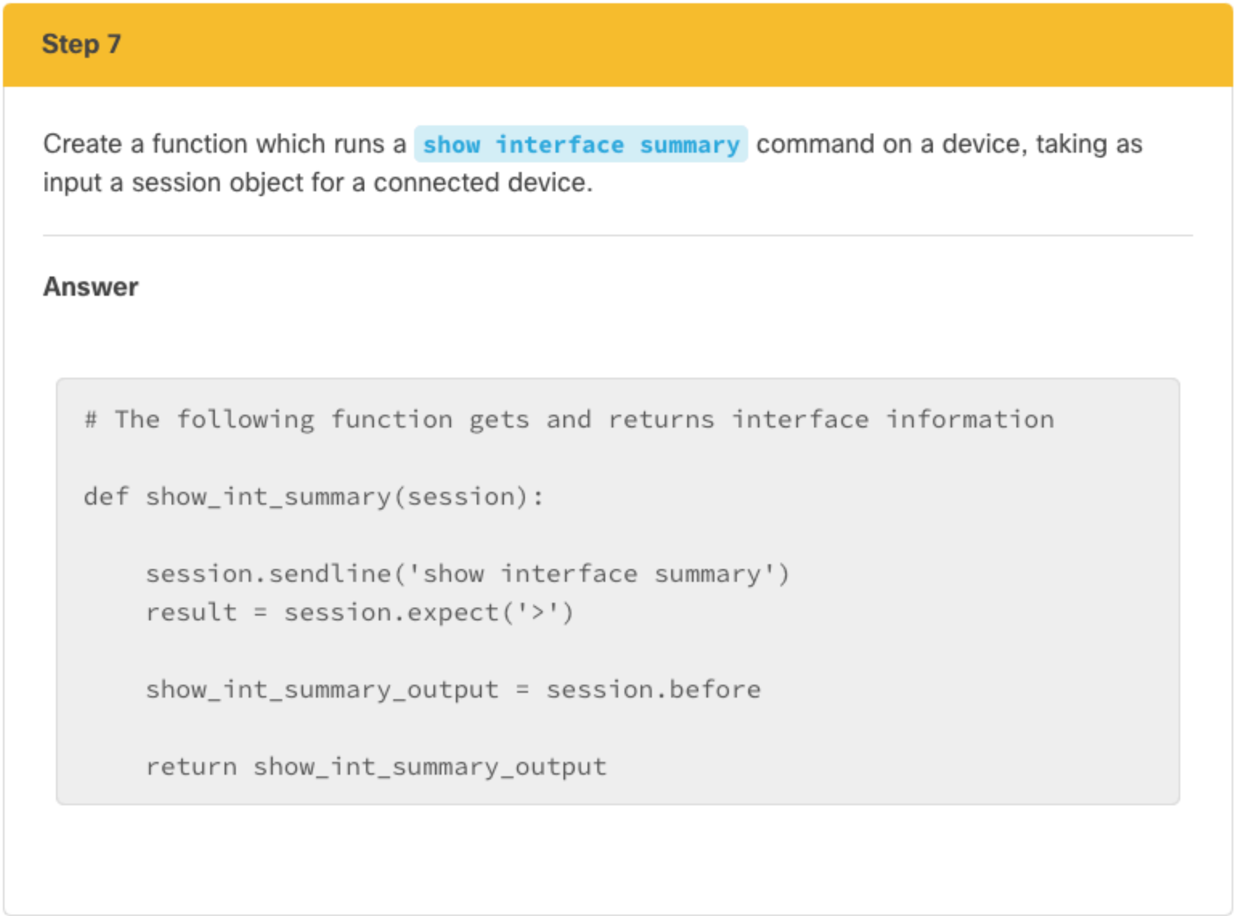
Answer

```
import pexpect

#-----------------------------------------------------------
def read_devices_info(filename):

    devices_list = []

    file = open(filename,'r')
    for line in file:

        device_info_list = line.strip().split(',')

        device_info = {}
        device_info['name'] = device_info_list[0]
        device_info['ip'] = device_info_list[1]
        device_info['username'] = device_info_list[2]
        device_info['password'] = device_info_list[3]

        devices_list.append(device_info)

    return devices_list
```

Step 6

Create a function to connect to a device, taking as parameters the IP address, username, and password for each device, and returning a Pexpect session.

```
# The following code connects to a device

def connect(dev_ip,username,password):

    print '--- connecting IOS: telnet '+dev_ip

    session = pexpect.spawn('telnet ' + dev_ip, timeout=20)

    result = session.expect(['Username:', pexpect.TIMEOUT])
    # Check for failure
    if result != 0:
        print '--- Timeout or unexpected reply from device'
        return 0

    print '--- attempting to: username: ' + username

    # Successfully got username prompt, logging with username
    session.sendline(username)

    result = session.expect(['Password:', pexpect.TIMEOUT])
    # Check for failure
    if result != 0:
        print '--- Timeout or unexpected reply from device'
        return 0

    print '--- attempting to: password: ' + password

    # Successfully got password prompt, logging in with password
    session.sendline(password)
    session.expect('>')

    return session  # return pexpect session object to caller
```

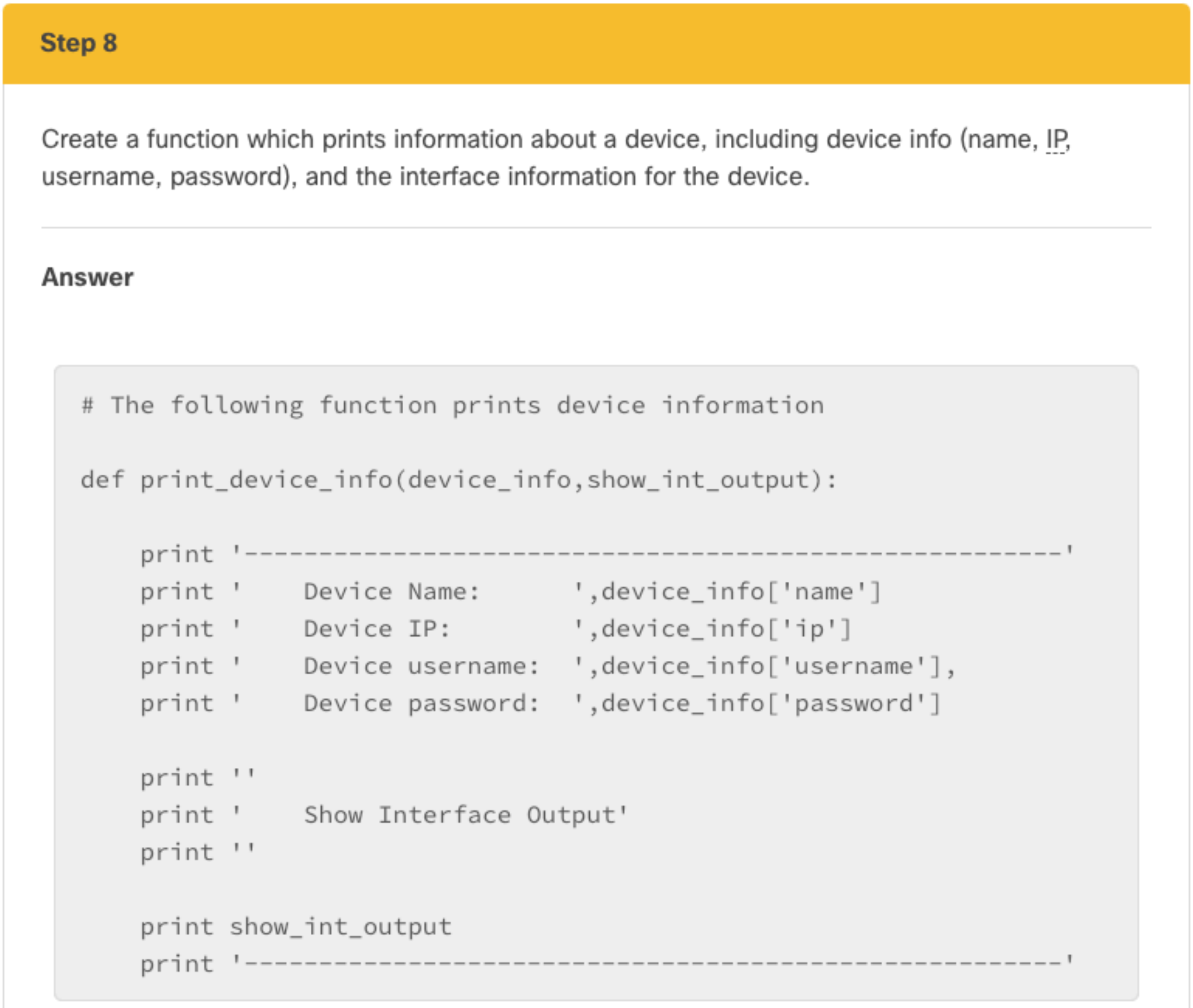


Step 7

Create a function which runs a `show interface summary` command on a device, taking as input a session object for a connected device.

Answer

```
# The following function gets and returns interface information

def show_int_summary(session):

    session.sendline('show interface summary')
    result = session.expect('>')

    show_int_summary_output = session.before

    return show_int_summary_output
```



Step 8

Create a function which prints information about a device, including device info (name, IP, username, password), and the interface information for the device.

Answer

```
# The following function prints device information

def print_device_info(device_info,show_int_output):

    print '-------------------------------------------------------'
    print '    Device Name:      ',device_info['name']
    print '    Device IP:        ',device_info['ip']
    print '    Device username:  ',device_info['username'],
    print '    Device password:  ',device_info['password']

    print ''
    print '    Show Interface Output'
    print ''

    print show_int_output
    print '-------------------------------------------------------'
```

Step 9

Create main application code which (a) reads device information from a file, (b) iterates through the list of devices, using your created functions to connect to each one, read interface information, and print nicely formatted output for each device.

Answer

```
# Main program: connect to device, show interface, display

if __name__ == '__main__':

    devices_list = read_devices_info('real-devices')

    for device_info in devices_list:

        session = connect(device_info['ip'],
                          device_info['username'],
                          device_info['password'])
        if session == 0:
            print '--- Session attempt unsuccessful ---'
            continue

        show_int_output = show_int_summary(session)

        print_device_info(device_info,show_int_output)

        session.sendline('quit')
        session.kill(0)
```