QA Tools Training Program:

Module-3 Session-1 Assessment

==========

1. Consider the following output of "show ap summary" of Controller 'ewlc'. Write parser for this output using RegEx and obtain the output in dictionary format.

ewlc#show ap summary

Number of APs: 3

CC = Country Code

RD = Regulatory Domain

AP Name Slots AP Model Ethernet MAC Radio MAC CC RD IP Address State Location

---

APBC26.C7A3.1970 2 AIR-AP3802E-B-K9 bc26.c7a3.1970 00b7.7166.bea0 US -B 192.165.7.199 Registered default location

APA4B2.3904.1F0C 3 C9130AXI-B a4b2.3904.1f0c 2c57.4156.9000 US -B 192.165.3.119 Registered default location

AP6849.92F9.8930 3 CW9163E-B 6849.92f9.8930 ecf4.0c4f.3360 US -B 192.165.8.139 Registered default location

Solution:

```python
#required module/function
import re

def parse_ap_summary(output):
    ap_dict = {}

    # pattern for matching
    pattern = re.compile(
        r"^(?P<AP_Name>\S+)\s+"
        r"(?P<Slots>\d+)\s+"
        r"(?P<AP_Model>\S+)\s+"
        r"(?P<Ethernet_MAC>\S+)\s+"
        r"(?P<Radio_MAC>\S+)\s+"
        r"(?P<CC>\S+)\s+"
        r"(?P<RD>\S+)\s+"
        r"(?P<IP_Address>\S+)\s+"
        r"(?P<State>\S+)\s+"
        r"(?P<Location>.+)$"
    )

    # Extract lines with AP information
    lines = output.splitlines()
    for line in lines:
        match = pattern.match(line)
        if match:
            ap_info = match.groupdict()
            ap_dict[ap_info['AP_Name']] = ap_info
```

```python
            ap_dict[ap_info['AP_Name']] = ap_info

    return ap_dict

# Sample output of "show ap summary"
output = """
ewlc#show ap summary
Number of APs: 3

CC = Country Code
RD = Regulatory Domain

AP Name                          Slots AP Model             Ethernet MAC   Radio MAC      CC   RD   IP Address             State       Location
-------------------------------------------------------------------------------------------------------------------------------------------------
APBC26.C7A3.1970                 2     AIR-AP3802E-B-K9     bc26.c7a3.1970 00b7.7166.bea0 US   -B   192.165.7.199          Registered  default location
APA4B2.3904.1F0C                 3     C9130AXI-B           a4b2.3904.1f0c 2c57.4156.9000 US   -B   192.165.3.119          Registered  default location
AP6849.92F9.8930                 3     CW9163E-B            6849.92f9.8930 ecf4.0c4f.3360 US   -B   192.165.8.139          Registered  default location
"""

# Parse the output
ap_summary_dict = parse_ap_summary(output)

# dictionary format output
import pprint
pprint.pprint(ap_summary_dict)
```

```
{'AP6849.92F9.8930': {'AP_Model': 'CW9163E-B',
                      'AP_Name': 'AP6849.92F9.8930',
                      'CC': 'US',
                      'Ethernet_MAC': '6849.92f9.8930',
                      'IP_Address': '192.165.8.139',
                      'Location': 'default location               ',
                      'RD': '-B',
                      'Radio_MAC': 'ecf4.0c4f.3360',
                      'Slots': '3',
                      'State': 'Registered'},
 'APA4B2.3904.1F0C': {'AP_Model': 'C9130AXI-B',
                      'AP_Name': 'APA4B2.3904.1F0C',
                      'CC': 'US',
                      'Ethernet_MAC': 'a4b2.3904.1f0c',
                      'IP_Address': '192.165.3.119',
                      'Location': 'default location               ',
                      'RD': '-B',
                      'Radio_MAC': '2c57.4156.9000',
                      'Slots': '3',
                      'State': 'Registered'},
 'APBC26.C7A3.1970': {'AP_Model': 'AIR-AP3802E-B-K9',
                      'AP_Name': 'APBC26.C7A3.1970',
                      'CC': 'US',
                      'Ethernet_MAC': 'bc26.c7a3.1970',
                      'IP_Address': '192.165.7.199',
                      'Location': 'default location               ',
                      'RD': '-B',
                      'Radio_MAC': '00b7.7166.bea0',
                      'Slots': '2',
                      'State': 'Registered'}}
```

✓ 0s completed at 8:32 PM

2. Using the parsed output of 1st question, write a script to perform the following:

- Fetch No of Slots, AP Model, Ethernet MAC, Radio MAC, CC, RD, IP Address of all the three AP's (APBC26.C7A3.1970, APA4B2.3904.1F0C and AP6849.92F9.8930) and print them.

- Verify 'State' for all the three AP's. Pass/Fail Criteria: Test should pass if state is 'Registered', otherwise it should fail.

Note:

Write the code for both questions in pyATS format:

- Create a separate function for parser

- Perform the operations of Question 2 in script file using the parsed output

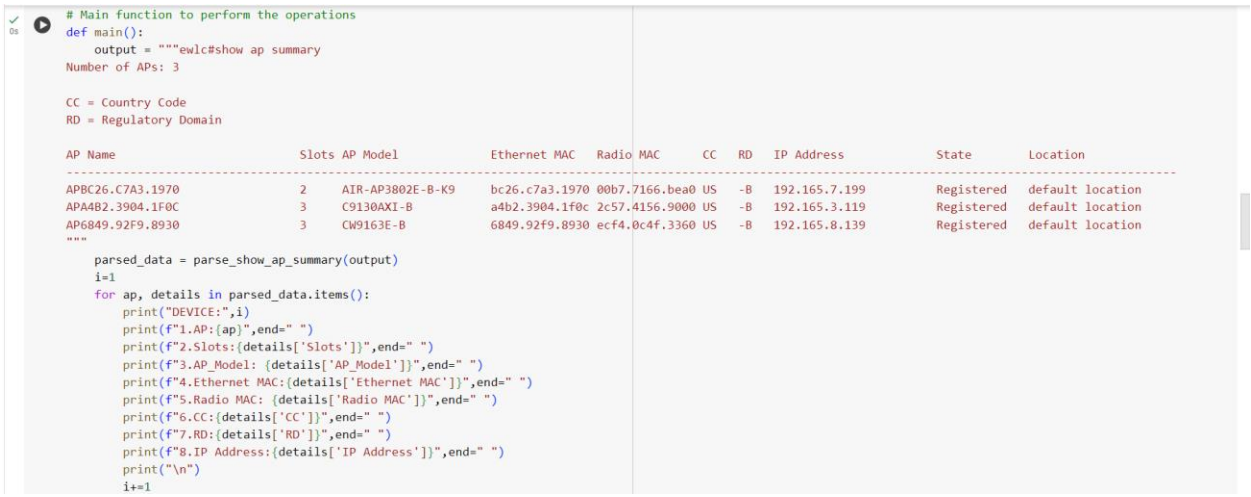
- Write a separate function for verifying AP state and use that in script

- Have the main function with run API in job file

Run it in any compiler to obtain the output.

```python
# Create the parser function
import re

def parse_show_ap_summary(output):
    data = {}
    lines = output.splitlines()
    pattern = re.compile(r'(\S+)\s+(\d+)\s+(\S+)\s+(\S+)\s+(\S+)\s+(\S+)\s+(\S+)\s+(\S+)\s+(\S+)\s+(\S+)\s+(\S+)\s+(\S+)\s+(\S+)\s+(\S+)\s+(.+)')
    for line in lines[5:]:  # Skip the header lines
        match = pattern.match(line)
        if match:
            data[match.group(1)] = {
                'Slots': match.group(2),
                'AP_Model': match.group(3),
                'Ethernet MAC': match.group(4),
                'Radio MAC': match.group(5),
                'CC': match.group(6),
                'RD': match.group(7),
                'IP Address': match.group(8),
                'State': match.group(9),
                'Location': match.group(10),
            }
    return data
```



```python
# Main function to perform the operations
def main():
    output = """ewlc#show ap summary
Number of APs: 3

CC = Country Code
RD = Regulatory Domain

AP Name                          Slots AP Model             Ethernet MAC   Radio MAC      CC   RD   IP Address            State       Location
-------------------------------------------------------------------------------------------------------------------------------------------------
APBC26.C7A3.1970                 2     AIR-AP3802E-B-K9     bc26.c7a3.1970 00b7.7166.bea0 US   -B   192.165.7.199         Registered  default location
APA4B2.3904.1F0C                 3     C9130AXI-B           a4b2.3904.1f0c 2c57.4156.9000 US   -B   192.165.3.119         Registered  default location
AP6849.92F9.8930                 3     CW9163E-B            6849.92f9.8930 ecf4.0c4f.3360 US   -B   192.165.8.139         Registered  default location
"""
    parsed_data = parse_show_ap_summary(output)
    i=1
    for ap, details in parsed_data.items():
        print("DEVICE:",i)
        print(f"1.AP:{ap}",end=" ")
        print(f"2.Slots:{details['Slots']}",end=" ")
        print(f"3.AP_Model: {details['AP_Model']}",end=" ")
        print(f"4.Ethernet MAC:{details['Ethernet MAC']}",end=" ")
        print(f"5.Radio MAC: {details['Radio MAC']}",end=" ")
        print(f"6.CC:{details['CC']}",end=" ")
        print(f"7.RD:{details['RD']}",end=" ")
        print(f"8.IP Address:{details['IP Address']}",end=" ")
        print("\n")
        i+=1
```

```
        print("VERIFICATION TEST\n")
        verify_ap_state(parsed_data)

    # Run the main function
    main()
```

```
DEVICE: 1
1.AP:APBC26.C7A3.1970 2.Slots:2 3.AP_Model: AIR-AP3802E-B-K9 4.Ethernet MAC:bc26.c7a3.1970 5.Radio MAC: 00b7.7166.bea0 6.CC:US 7.RD:-B 8.IP Address:192.165.7.199

DEVICE: 2
1.AP:APA4B2.3904.1F0C 2.Slots:3 3.AP_Model: C9130AXI-B 4.Ethernet MAC:a4b2.3904.1f0c 5.Radio MAC: 2c57.4156.9000 6.CC:US 7.RD:-B 8.IP Address:192.165.3.119

DEVICE: 3
1.AP:AP6849.92F9.8930 2.Slots:3 3.AP_Model: CW9163E-B 4.Ethernet MAC:6849.92f9.8930 5.Radio MAC: ecf4.0c4f.3360 6.CC:US 7.RD:-B 8.IP Address:192.165.8.139

VERIFICATION TEST

All AP states are verified successfully PASSED.
```