EX01: Network Command Practice

Task:
Perform network troubleshooting using the following commands:

ping, tracert, ipconfig, pathping, telnet, ftp, getmac, arp, hostname, nbtstat, netdiag, nslookup.

Instructions:

1. Use **ping** to test host reachability.
2. Use **tracert** to find the route to a host.
3. Use **ipconfig** to display IP configuration.
4. Use **pathping** to identify packet loss along the route.
5. Use **telnet** to test connectivity to a remote port.
6. Use **ftp** to connect to a server and list files.
7. Use **getmac** to find MAC addresses of interfaces.
8. Use **arp** to display IP-to-MAC mappings.
9. Use **hostname** to display the computer name.
10. Use **nbtstat** to check NetBIOS name tables.
11. Use **netdiag** to perform network diagnostics.
12. Use **nslookup** to resolve domain names.

Questions:
a) Show the command and its output for each task.
b) Explain the purpose of each command in **one line**.
c) Identify a scenario where each command is useful.

EX02: Switch Network Program (Short Version)

Task:

1. Connect **3 PCs (PC1, PC2, PC3)** to a switch.
2. Create VLANs:
   - VLAN 10 → PC1
   - VLAN 20 → PC2
   - VLAN 30 → PC3
3. Assign IPs:
   - PC1: 192.168.10.2/24
   - PC2: 192.168.20.2/24

    - PC3: 192.168.30.2/24

4. Verify VLANs and connectivity (ping).

Questions:
a) Draw network diagram.
b) Write switch commands to create VLANs and assign ports.
c) Explain why PCs in different VLANs cannot ping each other.

EX03: Static Router Configuration

Given:

- Router R1, R2, and R3 are connected.
- Networks:
    - R1 → 192.168.1.0/24
    - R2 → 192.168.2.0/24
    - R3 → 192.168.3.0/24

Task:

1. Configure **static routes** on each router so all networks can communicate.
2. Assign appropriate IP addresses to router interfaces.
3. Verify connectivity using **ping** between all networks.

Questions:
a) Write the static routing commands for each router.
b) Show ping results confirming connectivity.
c) Explain why static routing is needed in this scenario.

EX04: Client-Side API Interaction

Task:

1. Create an **HTML page** with a dropdown to select a student.
2. Use **jQuery AJAX** to fetch the list of students from:
    1. https://aec.sites.marrichet.com/api/student/
3. On selecting a student, fetch their result from:
    1. https://aec.sites.marrichet.com/api/result/?student__id=${studentId}
4. Display the **result details** on the page dynamically.

Questions:
a) Write the **HTML and jQuery code** to implement the task.
b) Explain how AJAX works in fetching and displaying API data.
c) Show a screenshot or console log of the API response for a selected student.

EX05: Client-Side GeoAPI Interaction

Task:

1. Create an **HTML page** and include **Leaflet via CDN**.
2. Use **AJAX** to fetch student locations from:
   1. https://aec.sites.marrichet.com/api/studentlocation/
3. Display each student as a **marker on the map** with popup showing their **name** and **location details**.
4. Center the map to show all markers.

Questions:
a) Write the HTML and JavaScript (Leaflet + AJAX) code.
b) Explain how markers are added dynamically from API data.
c) Show the map screenshot with markers for at least 3 students.

EX06: Networking and Data Transmission Task

Task: Choose **one of the following topics** and perform the required task:

1. **Data Link Layer Framing:** Implement **Character, Character Stuffing, and Bit Stuffing** methods.
2. **CRC Computation:** Compute **CRC-12, CRC-16, and CRC-CCITT** for a given data sequence.
3. **Sliding Window Protocol:** Simulate **Go-Back-N** and show frame transmission/retransmission.
4. **Dijkstra’s Algorithm:** Find the **shortest path** in a given network graph.
5. **Distance Vector Routing:** Determine **routing tables** for a given network.
6. **Leaky Bucket Algorithm:** Control **packet flow** using the leaky bucket method.
7. **Frame Sorting:** Sort frames in a **buffer** and display them in the correct sequence.

Questions for Submission:
a) Show your **implementation** (Python code, diagram, or table as applicable).
b) Explain the **steps or logic** used in your chosen task.
c) Show **sample output or results** for verification.

EX07: Data Encryption & Decryption – Custom Substitution Cipher

Task:

1. Implement **encryption** and **decryption** using a **custom substitution mapping**.
2. Given a mapping of letters (A-Z) to symbols/numbers, write Python functions to:
   - Encrypt a message using the mapping.
   - Decrypt the cipher back to the original message.

Questions:

a) Write the Python functions encrypt() and decrypt() using a dictionary mapping.
b) Encrypt the message: "HELLO WORLD" and show the output.
c) Decrypt the encrypted message and verify it matches the original.
d) Explain how the reverse mapping is used in decryption.

EX08: Web Application Using Web Server (Django)

Task:

1. Install Python and Django:
   1. python --version
   2. python -m pip install django
   3. python -m django --version
2. Create a new Django project named aecpro:
   1. python -m django startproject aecpro
   2. cd aecpro
3. Edit settings.py and update:
   1. ALLOWED_HOSTS = ["0.0.0.0", "*"]
4. Run the Django server on **port 8080** to access it over the network:
5. python manage.py runserver 0.0.0.0:8080

Questions:

a) Explain the purpose of ALLOWED_HOSTS in Django.
b) Show the steps and commands to start the Django server.
c) Access the web application from another computer in the same network and verify it runs.