Q1. Analyze a capture file to determine the most prevalent protocols and their proportional impact on the network.

- **Procedure:**

1. Perform a 5-minute Wireshark capture while using your computer normally (browsing, watching a video, etc.).
2. Stop the capture and navigate to **Statistics > Protocol Hierarchy**.
3. **Task:**
    - Identify the top three protocols by **Packets** percentage and report their values.
    - Identify the top three protocols by **Bytes** percentage and report their values.
    - Explain why there might be a difference between the top protocols by packet count versus byte count. For example, a protocol like DNS might have many packets but few bytes, while a video streaming protocol might have fewer packets but a high byte count.

Q2. Identify all unique devices (endpoints) on the network and analyze the conversations between them.

- **Procedure:**

1. Start a Wireshark capture and allow it to run for at least 5 minutes.
2. Stop the capture and go to **Statistics > Endpoints**.
3. Select the **IPv4** tab.
4. Go to **Statistics > Conversations**.
5. Select the **TCP** tab.
6. **Task:**
    - List the top five IP addresses found in the **Endpoints** table, sorted by **Packets** sent.
    - In the **Conversations** table, find the conversation with the highest number of packets. Report the two IP addresses involved and the total number of packets exchanged.
    - Based on the port numbers in that conversation (e.g., 80, 443, 21), infer the type of application or service being used. For example, port 443 suggests HTTPS traffic.

Q3.**Objective:** Use the I/O Graph to visualize network traffic and identify a significant traffic spike.

- **Procedure:**

1. Start a Wireshark capture.
2. Wait for the capture to stabilize, then perform an action that generates a lot of traffic, such as downloading a large file or starting a video stream.
3. Stop the capture after the activity is complete.
4. Navigate to **Statistics > I/O Graph**.
5. **Task:**
    - Take a screenshot of the I/O Graph showing the traffic spike.
    - Use the graph's timestamp to find the exact time of the traffic spike.

- Go back to the main Wireshark window and use a display filter like `frame.time_relative >= [time_of_spike]` to isolate the packets from that event.
- Report on the protocol that contributed the most to the traffic spike, using the **Protocol Hierarchy** feature on the filtered results.