Q1. You are building a student record management tool for a university system.

Requirements:

1. Define a structure Student with the following fields:
    - int id
    - string name
    - float gpa
2. Prompt the user to enter data for 5 students.
3. Write these records to a file named **students.txt** using **ofstream** with **ios::out** mode.
4. Then, allow the user to append 1 additional student record using **ios::app** mode (open in append mode and write without erasing old content).
5. Finally, read and display all student records from the file using **ifstream.**

Q2. A user wishes to update their resume in your Resume Builder application.

Requirements:

1. Prompt the user to enter:
    - string name
    - string email
    - int yearsOfExperience
    - string summary
2. Write this data to a file called **resume.txt.**
3. Use **ofstream** with **ios::trunc** to ensure that any previous contents are completely deleted before writing new data.
4. After writing, read back and display the contents to confirm that the file only has the latest entry.

Q3. You want to create a combined book by merging the contents of two chapters.

Requirements:

1. Assume two existing files: **chapter1.txt** and **chapter2.txt**. Populate them with some text before running this.
2. Open chapter1.txt and chapter2.txt using **ifstream** in **read mode**.
3. Open **book.txt** using **ofstream** in **ios::app** mode.
4. Read from both files and append the contents sequentially to **book.txt**.
5. Ensure a newline is added between chapter contents.
6. After writing, open and display book.txt contents to verify concatenation.

Q4. You are testing how file pointers work in a file while reading and writing.

Requirements:

1. Write the sentence "C++ is a powerful programming language." to a file named **info.txt** using **ofstream** in **ios::out** mode.
2. Reopen the file using **fstream** with both **ios::in | ios::out** modes.
3. Display the initial position of get pointer using **tellg()** and put pointer using **tellp()**.
4. Use **seekg(6)** to move the read pointer to the word "powerful" and read the word from there.
5. Use **seekp(6)** to move the write pointer to the same position, and overwrite "powerful" with "dynamic" (same length).
6. Use tellg() and tellp() again to show the new pointer positions.
7. Display the updated content of the file.

Q5. You're building a tool to search and replace a specific word in a text file *without rewriting the entire file*.

Requirements:

1. Create and populate a file named data.txt with a few sentences. (e.g., **AI is the future. AI will transform the world. Embrace AI now.**)
2. Prompt the user to enter:
    - A searchWord (e.g., "AI")
    - A replacementWord (e.g., "ML")
3. Open the file using **fstream** in **ios::in | ios::out** mode.
4. Search for the searchWord using **seekg()** and read sequentially.
5. When found:
    - Use **tellg()** to get the current read pointer location.
    - Use **seekp()** to position the write pointer.
    - Replace the word only if the replacement is of equal or smaller length to avoid shifting content.
6. If the word is replaced, pad with spaces if needed (e.g., replacing "AI" with "ML" or "XX").
7. After all replacements, display the updated file.