**🔷 THEORY**

**📌 Problem Statement**

You're tasked with building a system to manage student records stored in a **sequential file**. The operations supported are:

* **Add** a student
* **Delete** a student
* **Search** a student
* **Display all** records

Each student record contains:

* Roll number
* Name
* Division
* Address

**📌 Sequential File**

A **sequential file** stores records **one after another** in the order they are entered.

**Characteristics:**

* Simple structure, readable text format
* Good for **small-scale databases**
* Easy to implement with ifstream, ofstream, and fstream

**📌 File Operations Used:**

* ofstream with ios::app: Appends new records
* ifstream: Reads from file
* ofstream with ios::out: Rewrites the file (used in deletion)
* getline(): Reads lines from file
* find(): Searches substring in a line

**🔷 ALGORITHM**

**🔧 1. Add Student**

text

CopyEdit

1. Open "db.txt" in append mode.

2. Read roll number, name, division, and address from user.

3. Write data to file in formatted columns using setw().

4. Close the file.

**🔧 2. Delete Student**

text

CopyEdit

1. Open "db.txt" in read mode.

2. Read roll number to delete.

3. Read each line from file:

a. If the line does NOT contain the roll number → keep the line.

b. If the line contains the roll number → skip it (delete).

4. Close the read file.

5. Open "db.txt" in write mode.

6. Write back all kept lines.

7. Close the file and notify user if record was found or not.

**🔧 3. Search Student**

text

CopyEdit

1. Open "db.txt" in read mode.

2. Read roll number to search.

3. Read each line:

a. If roll number found in line → print the line, set found = true

4. If not found, display "Student not found".

5. Close the file.

**🔧 4. Display All Students**

text

CopyEdit

1. Open "db.txt" in read mode.

2. Read and print each line using getline().

3. Close the file.

**📌 Time Complexity (in terms of n = number of records):**

* Add: **O(1)** (append at end)
* Delete/Search: **O(n)** (must scan file linearly)
* Display All: **O(n)**