Problem Statement:1

You are required to create a MongoDB database named myDatabase with a collection called students. Implement the following tasks and queries to manage and retrieve data from the database. Use the example dataset provided below:

Example Dataset:

- Raj, age 22, subjects: ["Math", "Science"], graduated: false
- Priya, age 24, subjects: ["History", "Literature"], graduated: true
- Anil, age 23, subjects: ["Math", "Literature"], graduated: false
- Sita, age 25, subjects: ["Science", "Math"], graduated: true

Tasks and Queries:

1. **Inserting and Saving Documents:**
    - Perform a batch insert of the given student documents into the students collection.
    - Insert a single document for a new student, ensuring validation checks (e.g., name: "Reema", age: 21, subjects: ["Math", "Science"], graduated: false).
2. **Removing Documents:**
    - Remove the document for the student named "Raj" from the students collection.
    - Remove all documents for students who have not graduated.
3. **Updating Documents:**
    - Replace Priya's document with a new one that updates her subjects to ["History", "Philosophy"].
    - Update Anil's graduation status to true without replacing the entire document.
    - Perform an upsert operation to update Reema's document if it exists or insert it if it doesn't.
    - Update all documents to add a new field hasMath set to true for students who have "Math" in their subjects.
    - Return the updated document after changing Sita's graduation status to false.
4. **Executing Queries:**
    - Find and display all student documents in the students collection.
    - Find and display the document for the student named "Anil".
    - Retrieve all students older than 22 years.
    - Retrieve students whose name is either "Sita" or who are 23 years old.
    - Retrieve students who are not older than 23 years.
    - Retrieve students whose age is not 22.
    - Retrieve documents where the nickname field does not exist.

- Find students whose names end with the letter "a".
- Find students who have "Math" as one of their subjects.
- Execute a query using $where to find students older than 22 years.
- Limit the number of results to 2 when retrieving student documents.
- Skip the first document and retrieve the rest.
- Sort the student documents by age in ascending order.
- Perform an advanced query to find students older than 21, limit the results to 2, skip the first document, and sort the results by name in ascending order.

Problem Statement 2:

You are required to create a MongoDB database named `placementDB` with a collection called `placements`. Implement the following tasks and queries to manage and retrieve data from the database. Use the example dataset provided below:

Example Dataset:

- Raj, company: "TCS", designation: "Software Engineer", salary: 6, graduated: 2023
- Priya, company: "Infosys", designation: "Data Analyst", salary: 7, graduated: 2022
- Anil, company: "Wipro", designation: "System Engineer", salary: 5, graduated: 2023
- Sita, company: "HCL", designation: "Network Engineer", salary: 8, graduated: 2021

Tasks and Queries:

1. **Inserting and Saving Documents:**
   - Perform a batch insert of the given placement documents into the `placements` collection.
   - Insert a single document for a new placement, ensuring validation checks (e.g., name: "Reema", company: "Capgemini", designation: "Cyber Security Analyst", salary: 9, graduated: 2022).
2. **Removing Documents:**
   - Remove the document for the placement of "Raj" from the `placements` collection.
   - Remove all documents for placements with a salary less than 6 LPA.
3. **Updating Documents:**
   - Replace Priya's document with a new one that updates her designation to "Senior Data Analyst" and her salary to 8 LPA.
   - Update Anil's salary to 6 LPA without replacing the entire document.
   - Perform an upsert operation to update Reema's document if it exists or insert it if it doesn't.
   - Update all documents to add a new field `highSalary` set to true for placements with a salary above 7 LPA.
   - Return the updated document after changing Sita's company to "Cisco".
4. **Executing Queries:**

- Find and display all placement documents in the `placements` collection.
- Find and display the document for the placement of "Anil".
- Retrieve all placements with a salary greater than 6 LPA.
- Retrieve placements where the name is either "Sita" or the graduated year is 2023.
- Retrieve placements where the salary is not less than 7 LPA.
- Retrieve placements where the salary is not 6 LPA.
- Retrieve documents where the `department` field does not exist.
- Find placements where the company name ends with the letter "o".
- Find placements where the designation includes "Engineer".
- Execute a query using `$where` to find placements with a salary greater than 6 LPA.
- Limit the number of results to 2 when retrieving placement documents.
- Skip the first document and retrieve the rest.
- Sort the placement documents by salary in descending order.
- Perform an advanced query to find placements with a salary greater than 5 LPA, limit the results to 2, skip the first document, and sort the results by name in ascending order.

Problem Statement 3:

You are required to create a MongoDB database named `clubDB` with a collection called `studentClubs`. Implement the following tasks and queries to manage and retrieve data from the database. Use the example dataset provided below:

Example Dataset:

- Club: "Coding Club", president: "Raj", members: 50, established: 2019, active: true
- Club: "Literature Club", president: "Priya", members: 30, established: 2018, active: true
- Club: "Robotics Club", president: "Anil", members: 40, established: 2020, active: false
- Club: "Debate Club", president: "Sita", members: 20, established: 2021, active: true

Tasks and Queries:

1. **Inserting and Saving Documents:**
   - Perform a batch insert of the given club documents into the `studentClubs` collection.
   - Insert a single document for a new club, ensuring validation checks (e.g., club: "Photography Club", president: "Reema", members: 25, established: 2022, active: true).
2. **Removing Documents:**
   - Remove the document for the club named "Coding Club" from the `studentClubs` collection.
   - Remove all documents for clubs that are not active.
3. **Updating Documents:**

- Replace the document for "Literature Club" with a new one that updates the number of members to 35.
- Update the number of members in "Robotics Club" to 45 without replacing the entire document.
- Perform an upsert operation to update "Photography Club" if it exists or insert it if it doesn't.
- Update all documents to add a new field `majorEvent` set to true for clubs established before 2020.
- Return the updated document after changing the president of "Debate Club" to "Neha".

4. **Executing Queries:**
   - Find and display all club documents in the `studentClubs` collection.
   - Find and display the document for the club named "Robotics Club".
   - Retrieve all clubs with more than 30 members.
   - Retrieve clubs where the president is either "Sita" or the club was established in 2020.
   - Retrieve clubs that are active.
   - Retrieve clubs that were not established in 2019.
   - Retrieve documents where the `vicePresident` field does not exist.
   - Find clubs where the name ends with the letter "b".
   - Find clubs where the name includes "Club".
   - Execute a query using `$where` to find clubs with more than 20 members.
   - Limit the number of results to 2 when retrieving club documents.
   - Skip the first document and retrieve the rest.
   - Sort the club documents by the number of members in ascending order.
   - Perform an advanced query to find clubs with more than 25 members, limit the results to 2, skip the first document, and sort the results by name in ascending order.