

```
3, INSERT(4850407845c94b6a035e,A)
```

```
    oplog.mongo
```

```
1, INSERT(373807845c94b6a0335e,B)
```

```
2, READ(4850407845c94b6a035e)
```

```
3, INSERT(4850407845c94b6a035e,A)
```

Prepare a test case file that simulates a sequence of merge and operation commands. The test case should act as an input to your main program. A sample test case file could look like:

```
testcase.in
```

```
HIVE.INSERT(4850407845c94b6a035e,A)
```

```
HIVE.READ(4850407845c94b6a035e)
```

```
SQL.READ(373807845c94b6a0335e)
```

```
SQL.READ(4850407845c94b6a035e)
```

```
SQL.INSERT(4850407845c94b6a035e,A)
```

```
MONGO.INSERT(373807845c94b6a0335e,B)
```

```
MONGO.READ(4850407845c94b6a035e)
```

```
MONGO.INSERT(4850407845c94b6a035e,A)
```

```
MERGE(HIVE,SQL)
```

```
MERGE(SQL,HIVE)
```

```
MERGE(SQL,MONGO)
```

```
MERGE(MONGO,HIVE)
```

## Submission Guidelines

- The project submission deadline is on **April 29th, midnight**.
- By **April 30th**, each student group should meet with the TAs for a demo and evaluation based on mutual convenience. No extensions beyond **April 30th** will be allowed.
- Each student group should submit the source code along with a short report describing their approach.
- In the report, each student should explicitly self-declare their contribution out of 100. Failure to do so will indicate no or partial participation.
- We expect simple and feasible solutions rather than complex and non-intuitive ones.
- You may use any programming language of your choice for the implementation.
- The code should not be multi-threaded or involve parallel processing.
- Having a graphical user interface (GUI) is not mandatory.

Your final submission should include the source code, detailed explanations of the merge logic, the design of the operation log, sample logs and test cases, and a short write-up discussing the behavior and properties of your merge operations. The implementation must use at least three heterogeneous systems and must **include MongoDB**.