**Name: Ajinkya Kumbhar**

**Roll: BEAD20136**

**Subject: Computer Laboratory-IV BDA - 4**

**Class : BE**

**Branch: AI & DS**

**Title: Developing a MapReduce Program to Determine Student Grades**

**Problem Statement:** In educational institutions, especially in larger ones, managing and processing student data efficiently is crucial. One common task is determining the grades of students based on their performance in various subjects. To handle large datasets efficiently, parallel processing techniques like MapReduce can be employed. Therefore, the objective of this project is to develop a MapReduce program to compute the grades of students.

**Learning Objectives:**

1. Understand the concept of MapReduce and its application in parallel processing.
2. Gain proficiency in implementing MapReduce programs using a suitable framework (e.g., Hadoop, Apache Spark).
3. Learn to process large datasets efficiently for grade computation.
4. Enhance programming skills in Java or Python, depending on the chosen framework.
5. Explore methods for data validation and error handling within a MapReduce context.

**Procedure:**

1. **Understanding the Dataset:**
   * Begin by obtaining a dataset containing student information including their names, scores in different subjects, and other relevant details.
   * Ensure that the dataset is appropriately formatted and contains necessary information for grade computation.
2. **Designing the MapReduce Algorithm:**
   * Define the key-value pairs for the MapReduce algorithm. In this case, each student's name can be the key, and their scores in various subjects can be the values.
   * Implement the Map function to parse the input data and emit intermediate key-value pairs.
   * Implement the Reduce function to aggregate the scores and compute the grades for each student.
3. **Implementing the MapReduce Program:**
   * Depending on the chosen framework (e.g., Hadoop, Apache Spark), set up the development environment.
   * Write the MapReduce program in either Java or Python, adhering to the algorithm designed earlier.
   * Incorporate error handling and data validation mechanisms to ensure robustness.
4. **Testing and Debugging:**
   * Test the MapReduce program with sample datasets to verify its correctness and efficiency.
   * Debug any errors encountered during testing and refine the implementation accordingly.
5. **Scaling and Performance Optimization:**
   * Evaluate the performance of the MapReduce program with larger datasets.
   * Identify potential bottlenecks and optimize the program for better scalability and efficiency.
6. **Documentation and Reporting:**
   * Document the entire process, including the dataset used, algorithm design, implementation details, testing results, and performance analysis.
   * Prepare a comprehensive report summarizing the project, including the problem statement, learning objectives, procedure followed, challenges faced, and solutions devised.

**Conclusion:** In conclusion, the development of a MapReduce program for computing student grades offers significant advantages in terms of scalability and efficiency, especially when dealing with large datasets. Through this project, we have successfully achieved the objective of implementing a MapReduce algorithm to determine the grades of students based on their performance in various subjects. By adhering to best practices in algorithm design, programming, and performance optimization, we have demonstrated the potential of parallel processing techniques in educational data management. This project has not only enhanced our understanding of MapReduce but also equipped us with valuable skills in data processing and analysis.