Brief Description of How I Solved the Tasks

To solve the given tasks, I used the **MapReduce** programming model, which consists of two main functions: **Mapper** and **Reducer**. The tasks were divided as follows:

Task 1: Calculating Average Grade per Course

1. **Mapping:**
    - I created a function called map_course_grades() that iterates through each line of the dataset.
    - Each line is split into fields: year, course, grade, and university.
    - The function maps each course name to a list of grades associated with it.
2. **Reducing:**
    - The reduce_course_averages() function takes the mapped data and calculates the average grade for each course.
    - It sums the grades and divides by the number of grades to obtain the average.

Task 2: Calculating Average Grade per University

1. **Mapping:**
    - I created a function called map_university_grades() that maps university names to a list of grades.
    - Each line of the dataset is split into the same four fields as above.
2. **Reducing:**
    - The reduce_university_averages() function takes the mapped university grades and calculates the average grade for each university.

Bonus Task: Finding the Top 3 Grades per Year

1. **Mapping:**
    - I created a function called map_year_grades() to map each year to a list of grades.
    - The data is processed in a similar way by splitting each line.
2. **Reducing:**
    - The reduce_top_3_grades() function sorts the list of grades in descending order and selects the top three grades for each year.

Output and Execution

- The script uses an if __name__ == "__main__": block to execute the tasks.
- It prints the average grade per course, the average grade per university, and the top 3 grades per year in a well-formatted manner.

By dividing the tasks into mapping and reducing functions, the code becomes modular and efficient, following the MapReduce paradigm.