Tasks:
● Study Hadoop and MapReduce.
Link to slides:
https://docs.google.com/presentation/d/1Dv7_wv0MsvKyBs6tWWeMmRZn6jVPeRONWYHi4Q9fjL0/edit?usp=sharing

● Write the Mapper and Reducer routines (those necessary) to implement the queries mentioned in the following sections.
● Write the output-format routine to print results of queries to a text file. Write the code for these queries in python.

DataSet:
● ratings: UserID::MovieID::Rating::Timestamp
● users: UserID::Gender::Age::Occupation::Zip-code
● movies: MovieID::Title::Genres

Link to dataset: https://drive.google.com/open?id=1Gs7wLowvF6z_9QPmmtluPVByA_FPCbQI

Tutorial to solve a problem using map-reduce:
Map Reduce is just a different way of solving problems. Let us start with a simple word count problem. Say, we have a text file and we want to count the frequency of occurrence of each word. The tutorial below explains how to solve this problem using a map-reduce algorithm.
http://www.michael-noll.com/tutorials/writing-an-hadoop-mapreduce-program-in-python/

Queries:
The queries are to be implemented in the Mapper and Reducer phases. Some of them may be empty based on the query. Here are some simple queries to get you started. You need to implement these following queries in the assignment.

❖ Query 1: Find all the unique user ids who have rated at least 276 movies.
❖ Query 2: Find the year when the maximum number of 'Action' movies are released.
❖ Query 3: List all the users whose average ratings are greater than 3.
❖ Query 4: Name the movie which got the best average ratings considering that minimum 10 users should rate that movie.
❖ Query 5: List all the occupation along with the number of users in each occupation.
❖ Query 6: Given an input zip code, find all the user-ids that belongs to that zip code. You must take the input zip code through the command line.

Logic:
Hadoop has the following phases:
1. map. // Execute operation on each record.
2. combiner. // We do not need to code this.
3. partitioner. // We do not need to code this.
4. reduce. // Combine the results from the combiner.
5. Output format.
How to run and test your code:
Since we do not have access to a Hadoop cluster, we will be testing our codes on a Linux system as follows:
cat input.txt | python mapper.py | sort | python reducer.py

Explanation of the above commands:
● “cat” is a Linux command to print the contents of a file on the console.
● You must be familiar with the pipe operator (|). It directs the output of the previous command to the next command.
● “sort” is a Linux command to sort the input lexicographically.

Deliverables:
Python codes for the above functionality compressed as .tar.gz, named <your-name>.tar.gz.