

**Gauranga Kumar Baishya**  
**Chennai Mathematical Institute**  
**MDS202325**  
**Assignment 4**

**Answer to Question 1(a)**

```
my_chess_data = LOAD 'C:/games.csv'
USING PigStorage(',')
AS (
    id:chararray,
    rated:boolean,
    created_at:long,
    last_move_at:long,
    turns:int,
    victory_status:chararray,
    winner:chararray,
    increment_code:chararray,
    white_id:chararray,
    white_rating:int,
    black_id:chararray,
    black_rating:int,
    moves:chararray,
    opening_eco:chararray,
    opening_name:chararray,
    opening_ply:int
);
winner = FILTER chess_games BY winner=='white' ;

STORE winner INTO 'C:/white_winner'
USING PigStorage(',');
```

**Answer to Question 1(b)**

```
Groupd = GROUP winner ALL;

my_avg = FOREACH Groupd GENERATE AVG(winner .white_rating);

DUMP my_avg;
```

**Answer to Question 1(c)**

```
_100_turns = FILTER my_games_data BY turns>100;

group_of_100_turns = GROUP _100_turns ALL;

my_count = FOREACH group_of_100_turns GENERATE
COUNT(_100_turns);
```

```
DUMP my_count;
```

## Answer to Question 2

### Map-Reduce Workflow for Word Length Frequency

#### 1: Mapping

- **Input:** Large text document
- **Output:** Key-value pairs where each word's length is the key and the value is 1.

Word  $x \rightarrow \langle \text{length of } x, 1 \rangle$

**2: Shuffle and Sort** Key-value pairs are sorted by their keys and sent to the reducer.

**3: Reduce** For each unique key, aggregate the key-value pairs to find the frequency of word lengths. Each key-value pair represents the word length and its frequency.

$\langle \text{word length, frequency} \rangle$

**Example:** "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua."

Mapper		Shuffle and Sort		
'Lorem'	<5, 1>	<2, 1>		
'ipsum'	<5, 1>	<2, 1>		
'dolor'	<5, 1>	<2, 1>		
'sit'	<3, 1>	<3, 1>		
'amet,'	<4, 1>	<3, 1>		Reduce
'consectetur'	<11, 1>	<4, 1>		
'adipiscing'	<10, 1>	<4, 1>		<2, 3>
'elit.'	<4, 1>	<5, 1>		<3, 2>
'Sed'	<3, 1>	<5, 1>		<4, 2>
'do'	<2, 1>	<5, 1>		<5, 4>
'eiusmod'	<7, 1>	<5, 1>		<6, 4>
'tempor'	<6, 1>	<6, 1>		<7, 1>
'incididunt'	<10, 1>	<6, 1>		<10, 2>
'ut'	<2, 1>	<6, 1>		<11, 1>
'labore'	<6, 1>	<6, 1>		
'et'	<2, 1>	<7, 1>		
'dolore'	<6, 1>	<10, 1>		
'magna'	<5, 1>	<10, 1>		
'aliqua.'	<6, 1>	<11, 1>		

#### Result:

2, 3  
3, 2  
4, 2  
5, 4

6, 4  
7, 1  
10, 2  
11, 1