Tasks

1. **Read the data. (10%)**
    1. Read in the data using the mongoimport command.

       Hint: The data is in the form of an array of JSON objects. So the mongoimport command you have used in the module will not work directly. You will have to let mongoimport know that the file you are importing is not just a JSON file but an array of JSON files. The syntax mongoimport for the JSON array is

       ```
       mongoimport <usual mongo import command> -- jsonArray
       ```

    2. Display data for the match with “game” = “3”

       Suggestion: Use the pretty function to make the data readable.

2. **Explore the data for game 3. (40%)**

    In this subtask, you are expected to create a new collection with only the document relating to game 3. You can create the new document with the help of the code below.

    ```
    db.<new_collection_name>.save( <query that you used to display data for match 3>.toArray()  )
    ```

    1. Display the Game Feed data for the game in the new collection.

    2. Display the last event in game 3.
       Hint: Event is a nested field. You can use the dot operator to access the nested fields. For example, to access the Action field inside the Game Feed, you can write the query as

       ```
       {"$Game_Feed.Action": <condition>}
       ```

    3. Who won game 3, imposters or crew?
       How will you know who won the game? check the 'Game Feed' for all the events in game 3.
       Write a query to display only the Game Feed field for game 3. Note the lack of underscore between Game and Feed.

This field is a nested field inside the Game_Feed field. Refer to serial number 8 in the Game_Feed data dictionary.

4. Who picked the black color in game 3? Was that player crew or imposter?

5. How many voting events happened in game 3?

6. This task was focused on exploring a single document. As explained earlier the document is made of 4 main fields and three of those are nested and hold a lot of data.
   If you were to redesign this database to make it easier to query what changes would you make to the structure? Explain your design decisions. Do not modify the structure, simply explain the changes you want to make and justify them.

3. **Overall aggregation. (40%)**

   For this set of questions, use the earlier data collection with all 499 games.

   1. How many events in total do you have data for, in this collection (across all games)?
      An event as explained earlier is a record of development in the game. For instance, game 3 had 14 events.
   2. How many matches did the crew win versus how many matches did the impostors win?
      Hint: Use regular expressions to analyze the game feed column for all entries documents where the Outcome field is not empty.
   3. Find out how many matches were played on each map. Your output should contain the names of maps and the number of games played on each map.
      Hint: You will have to ensure only one event from each game is filtered before grouping.
   4. How many times in total across all games did the crew skip a vote?
   5. How many times in total across all matches does the crew vote against imposters?

   6. The questions you answered in this task were all related to high-level aggregations across the entire collection. In your opinion, is the game more or less hard for impostors? Justify your answer with suitable insights from the data.

4. **Player-level aggregation. (10%)**

   In this set of tasks, we will aggregate the data player-wise. Let's say that we want to create a dashboard to select players for games. You have to convert the data to a structured format where each row represents one player and some of their statistics.

1. Find the number of unique players in the data set.

   Hint: Check the data type of the output of the query you write to get the set of unique players, and then decide how you will calculate the number of unique players.
2. Who is the best crew member?

   Find the player who (as a crewmate) voted to remove the imposters the most number of times.
3. Who is the worst crew member?

   Find the player who (as a crewmate) voted to remove the other crew members the most number of times.
4. Find the win percentage for all players. *(Optional)*

   Win percentage can be calculated by the total number of games the player has won divided by the total matches the player has participated in.
5. Find the color preference of all players. *(Optional)*

   The color preference of a player is the color they picked the most number of times.

5. **Create an export from MongoDB in the form given below (Optional)**

   As discussed earlier the aim of this exercise was to create a table for predictive and descriptive analysis. The table given below is one way to create a structured output from the given data.

Player name	**Games won as imposter**	**Games won as crew**	**Win percentage (overall)**	**Voted Against Imposter**	**Voted against Crew members**	**Color preference**	**Voting rate**