Movie Rating System

Loops

1. **Calculate Average Ratings**

   - **Task:** Write a function that calculates the average rating for each movie in the given array and returns an array with the average ratings.
   - **Inputs:** An array of movie objects.
   - **Example:**
     - Input:
       ```
       [{ title: "Movie 1", ratings: [8, 7, 9]}, { title: "Movie 2", ratings: [6, 8, 7]}, { title: "Movie 3", ratings: [9, 9, 10]}]
       ```
     - Output: `[8, 7, 9.33]`
   - **Tip:** Use nested loops to iterate over the movies and their ratings, summing up the ratings for each movie and calculating the average.

2. **Find Top Rated Movies**

   - **Task:** Write a function that finds the movies with the highest ratings in the given array and returns an array with their titles.
   - **Inputs:** An array of movie objects.
   - **Example:**
     - Input:
       ```
       [{ title: "Movie 1", rating: 8 }, { title: "Movie 2", rating: 7 }, { title: "Movie 3", rating: 9 }]
       ```
     - Output: `["Movie 3"]`
   - **Tip:** Implement a loop to compare the ratings of each movie and keep track of the top-rated movies.

3. **Create Rating Matrix**

   - **Task:** Write a function that creates a 2D array (matrix) where each row represents a movie and each column represents a rating.
   - **Inputs:** An array of movie objects.
   - **Example:**

    - Input:
      `[{ title: "Movie 1", ratings: [8, 7, 9]}, { title: "Movie 2", ratings: [6, 8, 7]}]`
    - Output: `[[8, 7, 9], [6, 8, 7]]`
  - **Tip:** Use loops to populate the matrix with the ratings of each movie.

4. **Count Movies with a Rating Above Threshold**

  - **Task:** Write a function that counts the number of movies in the given array that have a rating above a specified threshold.
  - **Inputs:** An array of movie objects, a rating threshold.
  - **Example:**
    - Input:
      `[{ title: "Movie 1", rating: 8 }, { title: "Movie 2", rating: 7 }, { title: "Movie 3", rating: 9 }]`
      , rating threshold: 8
    - Output: `2`
  - **Tip:** Use a while loop to iterate over the movies, increment a counter variable for each movie with a rating above the threshold, and return the final count.

5. **Find First Movie with a Specific Genre**

  - **Task:** Write a function that finds the first movie in the given array that has a specific genre.
  - **Inputs:** An array of movie objects, a genre.
  - **Example:**
    - Input:
      `[{ title: "Movie 1", genres: ["Action", "Drama"] }, { title: "Movie 2", genres: ["Drama", "Romance"] }, { title: "Movie 3", genres: ["Action", "Thriller"] }]`
      , genre: "Romance"
    - Output: `{ title: "Movie 2", genres: ["Drama", "Romance"] }`
  - **Tip:** Use a while loop to iterate over the movies, check if the current movie has the specified genre, and return the first movie that matches the genre.

6. **Draw Movie Rating Chart**

  - **Task:** Write a function that draws a chart or graph representation of the movie ratings using loops.

- **Inputs:** An array of movie objects.
- **Example:**
  - Input:
    ```
    [{ title: "Movie 1", rating: 8 }, { title: "Movie 2", rating: 7 }, { title: "Movie 3", rating: 9 }]
    ```
  - Output:

    ```
    Movie 1: ********
    Movie 2: *******
    Movie 3: *********
    ```

- **Tip:** Use loops to iterate over the movies and their ratings to visualize the ratings in a meaningful way.

Bonus

7. **Calculate Average Rating for Each Movie**

   - **Task:** Write a function that calculates the average rating for each movie and returns an array with the average ratings.
   - **Inputs:** An array of movie objects.
   - **Example:**
     - Input:
       ```
       [{ title: "Movie 1", ratings: [8, 7, 9]}, { title: "Movie 2", ratings: [6, 8, 7]}]
       ```
     - Output: `[8, 7]`
   - **Tip:** Use nested for loops to iterate over the movies and their ratings, calculate the average for each movie, and store the results in an array.

8. **Find Movies with the Highest Rating in Each Column**

   - **Task:** Write a function that finds the movies with the highest rating in each column of the rating matrix and returns an array with their titles.
   - **Inputs:** An array of movie objects.
   - **Example:**
     - Input:
       ```
       [{ title: "Movie 1", ratings: [8, 7, 9]}, { title: "Movie 2", ratings: [6, 8, 7]}]
       ```
     - Output: `["Movie 1", "Movie 2", "Movie 1"]`

- **Tip:** Use nested for loops to iterate over the columns of the rating matrix and find the movies with the highest rating in each column.