Homework 07:

```
 /*
    Given
          https://reqres.in/api/unknown/
    When
       I send GET Request to the URL
    Then

       1)Status code is 200
       2)Print all pantone_values
       3)Print all ids greater than 3 on the console
         Assert that there are 3 ids greater than 3
       4)Print all names whose ids are less than 3 on the console
         Assert that the number of names whose ids are less than 3 is 2
  */
```

"Homework 07," which requires sending a GET request to an API endpoint and validating the response, along with specific criteria on the data returned, you can implement this using JavaScript with Node.js and the Axios library. This method enables both printing certain data to the console and asserting specific conditions programmatically. This approach is suitable for integrating into automated testing suites or for backend development testing practices.

Using JavaScript and Axios for Programmatic API Testing

1. **Prepare Your Environment**:
    - Ensure Node.js is installed on your computer.
    - Create a directory, initialize a Node.js project, and install Axios:

```
mkdir my-api-test cd my-api-test npm init -y npm install axios
```

2. **Write the Test Script**:
    - Create a file named **test.js**.

- Add the following JavaScript code to perform the GET request, print the required data, and assert conditions based on the homework requirements:

```
const axios = require('axios'); axios.get('https://reqres.in/api/unknown/') .then(response => {
console.log("HTTP Status Code check:", response.status === 200); // Status code 200 check // Print all
pantone_values console.log("Pantone Values:"); response.data.data.forEach(item =>
console.log(item.pantone_value)); // Print all IDs greater than 3 const idsGreaterThanThree =
response.data.data.filter(item => item.id > 3).map(item => item.id); console.log("IDs greater than 3:",
idsGreaterThanThree); console.assert(idsGreaterThanThree.length === 3, `Expected 3 IDs greater than 3,
found ${idsGreaterThanThree.length}`); // Print all names whose IDs are less than 3 const
namesWithIdsLessThanThree = response.data.data.filter(item => item.id < 3).map(item => item.name);
console.log("Names with IDs less than 3:", namesWithIdsLessThanThree);
console.assert(namesWithIdsLessThanThree.length === 2, `Expected 2 names with IDs less than 3, found
${namesWithIdsLessThanThree.length}`); }) .catch(error => { console.error('Error during API call', error);
});
```

3. **Execute the Test**:

    - Run the script by typing in your terminal:

```
node test.js
```

This script makes a GET request to the provided URL and performs several operations:

- It checks that the HTTP status code is 200.
- It prints all 'pantone_values' from the response.
- It finds all IDs greater than 3, prints them, and asserts that exactly three such IDs exist.
- It finds and prints all names where the ID is less than 3, and asserts that there are exactly two such names.

This approach efficiently satisfies the requirements of Homework 07, providing a clear and concise way to validate API responses and handle data assertions programmatically. This setup is ideal for automated testing environments where such validations are part of continuous integration testing cycles.