System Requirements Specification Index

For

Multidimensional Array

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

Table of Contents

1 PROJECT ABSTRACT

This project assesses knowledge of Java multi-dimensional arrays (2D arrays) and basic operations performed on them.

The tasks involve iterating over 2D arrays to print elements, calculate row lengths, find maximum values, sum elements of specific rows, and transpose the arrays.

2 ASSESSMENT TASKS

Task 1: Print Elements of a 2D Array Using Nested For Loops:

- Print the message:
  `"Task 1: Print Elements of a 2D Array"`.
- Declare and initialize a 2D integer array `arr1` with the elements:
  ```
  {
   {1, 2, 3},
   {4, 5, 6},
   {7, 8, 9}
  }
  ```
- Use an outer `for` loop to iterate through the rows of `arr1`:
  - ➔ Use an inner `for` loop to iterate through the columns of each row.
  - ➔ Print each element followed by a space.
  - ➔ After each row, print a new line using `System.out.println()`.
- This prints the full 2D array in matrix form.

Expected Output:

```
Task 1: Print Elements of a 2D Array
1 2 3
4 5 6
7 8 9
```

Task 2: Find the Length of a Row in a 2D Array:

- Print the message:
  `"Task 2: Find the Length of a Row in a 2D Array"`.
- Declare and initialize a 2D integer array `arr2` with the same elements as above.
- Use `arr2[0].length` to get the length of the first row.
- Print the result:
  `"Length of row 1: <length>"`.
- This will display how many columns are in the first row.

Expected Output:

```
Task 2: Find the Length of a Row in a 2D Array
Length of row 1: 3
```

Task 3: Find the Maximum Value in a 2D Array Using Nested For Loops:

- Print the message:
  `"Task 3: Find the Maximum Value in a 2D Array"`.
- Declare and initialize a 2D integer array `arr3` with the same elements.
- Declare an integer variable `max` and initialize it with the first element `arr3[0][0]`.
- Use an outer `for` loop to iterate through the rows:
  - ➔ Use an inner `for` loop to iterate through the columns of each row.
  - ➔ In each iteration, check if the current element is greater than `max`:
    - If true, update `max` with that element.
- After the loop, print:
  `"Maximum Value: <max>"`.
- This will print the largest value in the array.

Expected Output:

```
Task 3: Find the Maximum Value in a 2D Array
Maximum Value: 9
```

Task 4: Calculate Sum of Elements in a Row of a 2D Array:

- Print the message:
  `"Task 4: Sum of Elements in a Row of a 2D Array"`.
- Declare and initialize a 2D integer array `arr4` with the same elements.
- Declare an integer variable `sum` and initialize it to `0`.
- Use a single `for` loop to iterate through the second row (`arr4[1]`):
  - ➔ Add each element to `sum`.
- After the loop completes, print:
  `"Sum of row 2 elements: <sum>"`.
- This will display the sum of elements in the second row.

Expected Output:

```
Task 4: Sum of Elements in a Row of a 2D Array
Sum of row 2 elements: 15
```

Task 5: Transpose a 2D Array Using Nested For Loops:

- Print the message:
  `"Task 5: Transpose a 2D Array"`.
- Declare and initialize a 2D integer array `arr5` with elements:
  ```
  {
   {1, 2, 3},
   {4, 5, 6}
  }
  ```
- Declare a new 2D integer array `transposed` with dimensions

[arr5[0].length][arr5.length].

- Use an outer for loop to iterate through the rows of arr5:
  - ➔ Use an inner for loop to iterate through the columns.
  - ➔ Assign arr5[i][j] to transposed[j][i].
- After populating transposed, print:
  "Transposed Array:".
- Use nested for loops to print the elements of transposed in matrix form.

Expected Output:

```
Task 5: Transpose a 2D Array
Transposed Array:
1 4
2 5
3 6
```

3 Template Code Structure

3.1 Package: com.yaksha.assignment.MultiArrayAssignment

Resources

Class/Interface	Description	Status
MultiArrayAssignment (class)	● Main class demonstrating basic operations on **multi-dimensional (2D) arrays**. ● Includes examples of: - Printing elements of a 2D array. - Finding the length of a row in a 2D array. - Finding the maximum value in a 2D array. - Summing elements in a row. - Transposing a 2D array.	Need to be implemented.

4 Execution Steps to Follow

1. **All actions like build, compile, running application, running test cases will be through Command Terminal.**
2. **To open the command terminal the test takers, need to go to Application menu (Three horizontal lines at left top) Terminal New Terminal.**
3. **This editor Auto Saves the code.**
4. **If you want to exit(logout) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use CTRL+Shift+B-command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.**
5. **These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.**
6. **To run your project use command:**

   ```
   mvn compile exec:java
   -Dexec.mainClass="com.yaksha.assignment.MultiArrayAssignment"
   ```

7. **To test your project test cases, use the command**

   ```
   mvn test
   ```

8. **You need to use CTRL+Shift+B - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.**