

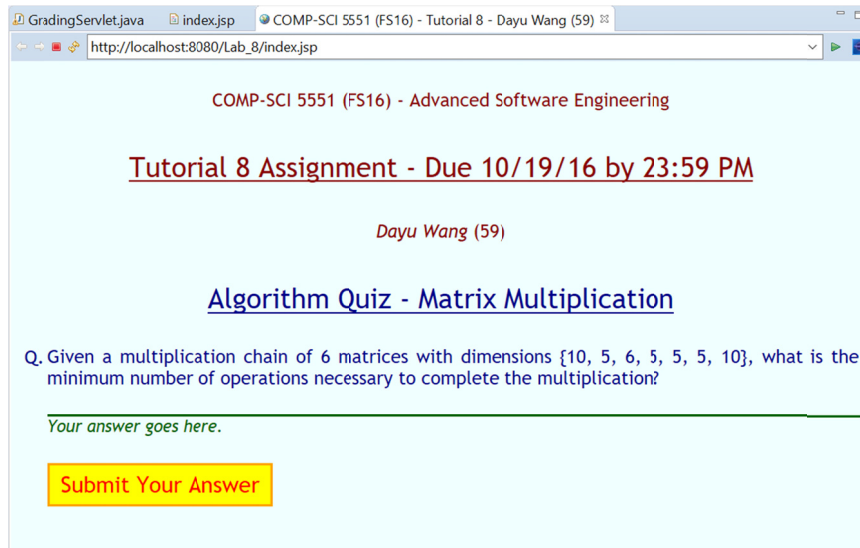
COMP-SCI 5551 (FS16) - Advanced Software Engineering

Tutorial 8 Assignment (Due 10/19/16 by 23:59 PM)

Dayu Wang (59)

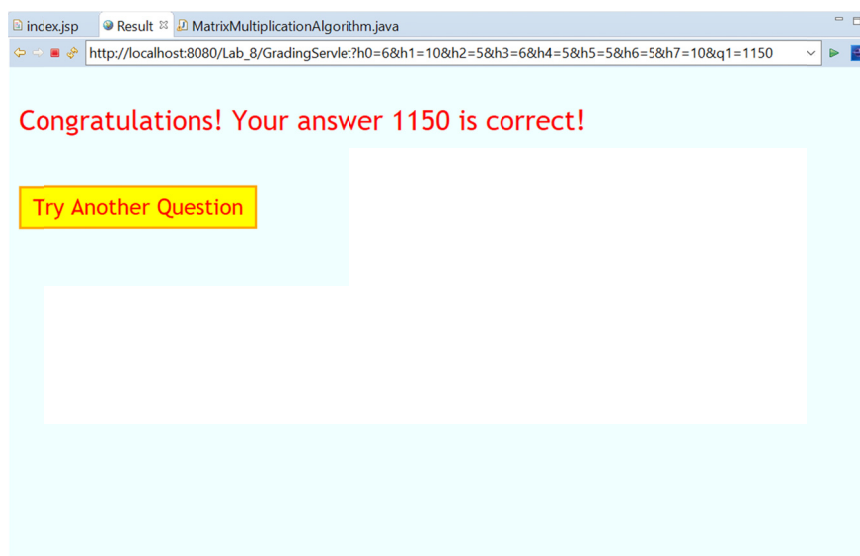
Screenshots

1. Main JSP Page

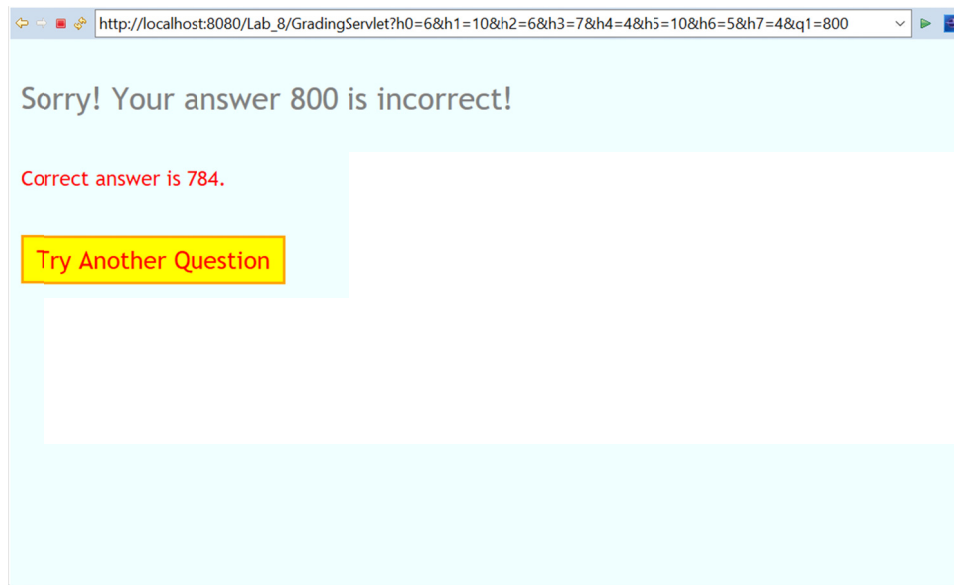


The number of matrices and the dimension array are **randomly** generated each time when we load the web page.

2. If the answer is correct...



3. If the answer is NOT correct...



4. Junit Test Cases

The screenshot shows an IDE with the file `MatrixMultiplicationAlgorithmTest.java` open. The code defines a class with two test methods: `testSolution()` and `testMain()`. The `testSolution()` method creates a `MatrixMultiplicationAlgorithm` instance, uses a predefined test case, and asserts the result. The `testMain()` method prints the test case and result, then asserts the result. The JUnit runner shows that both tests passed successfully after 0.035 seconds.

```
1 package com.cs5551.fs16.DayuWang.tutorial8;
2
3 import static org.junit.Assert.*;
4
5 public class MatrixMultiplicationAlgorithmTest {
6
7     @Test
8     public void testSolution() {
9         MatrixMultiplicationAlgorithm chain = new MatrixMultiplicationAlgorithm();
10        int[] testCase = {7, 10, 5, 16, 9, 22};
11        int realAnswer = chain.Solution(testCase);
12        int testRealAnswer = 2771;
13    }
14
15    @Test
16    public void testMain() {
17        int[] testCase = {7, 10, 5, 16, 9, 22};
18        System.out.println("@Test solution(): " + Arrays.toString(testCase));
19        int result = new MatrixMultiplicationAlgorithm().Solution(testCase);
20        System.out.println("@result: " + " = " + "2771");
21        assertEquals(result, 2771);
22    }
23 }
24
25
26
27
28
29
```

Finished after 0.035 seconds

Runs: 2/2 Errors: 0 Failures: 0

> com.cs5551.fs16.DayuWang.tutorial8.MatrixMultiplicationAlgorithmTest [Runner: JUnit 4] (0.001 s) Failure Trace

The screenshot shows an IDE with the file `RandomNumberTest.java` open. The code defines a class with two test methods: `testRandomNumber()` and `testGenerate()`. The `testRandomNumber()` method creates a `RandomNumber` instance with parameters 3 and 6, generates a random number, and asserts it falls within the range [3, 6]. The `testGenerate()` method does the same but also asserts the generated number is greater than or equal to the lower bound. The JUnit runner shows that both tests passed successfully after 0.026 seconds.

```
1 package com.cs5551.fs16.DayuWang.tutorial8;
2
3 import static org.junit.Assert.*;
4
5 public class RandomNumberTest {
6
7     @Test
8     public void testRandomNumber() {
9         int down = 3;
10        int up = 6;
11        RandomNumber r = new RandomNumber(3, 6);
12        int real = r.Generate();
13        System.out.println(down + " <= " + real + " <= " + up);
14    }
15
16    @Test
17    public void testGenerate() {
18        int down = 3;
19        int up = 6;
20        RandomNumber r = new RandomNumber(3, 6);
21        int real = r.Generate();
22        System.out.println(down + " <= " + real + " <= " + up);
23        assertEquals((down <= real), true);
24        assertEquals((up >= real), true);
25    }
26 }
27
28
29
30
```

Finished after 0.026 seconds

Runs: 2/2 Errors: 0 Failures: 0

> com.cs5551.fs16.DayuWang.tutorial8.RandomNumberTest [Runner: JUnit 4] (0.000 s) Failure Trace