**Test of connected graph**

**1. Test pass but graph is not connected**

import org.junit.After;

import org.junit.AfterClass;

import org.junit.Before;

import org.junit.BeforeClass;

import org.junit.Test;

import static org.junit.Assert.\*;

public class graphConnectedTest {

public graphConnectedTest() {

}

public static void setUpClass() {

}

public static void tearDownClass() {

}

@Before

public void setUp() {

}

public void tearDown() {

}

public void testBfs() {

System.out.println("bfs");

int[][] adjacency\_matrix = {{0,1,0},{1,0,0},{0,0,0}};

int source = 1;

graphConnected instance = new graphConnected();

String testresult=instance.bfs(adjacency\_matrix, source);

String result1="is connected";

String result2="not connected";

if(result1.equals(testresult))

{

System.out.println("pass");

} else if(result2.equals(testresult)){

System.out.println("pass");

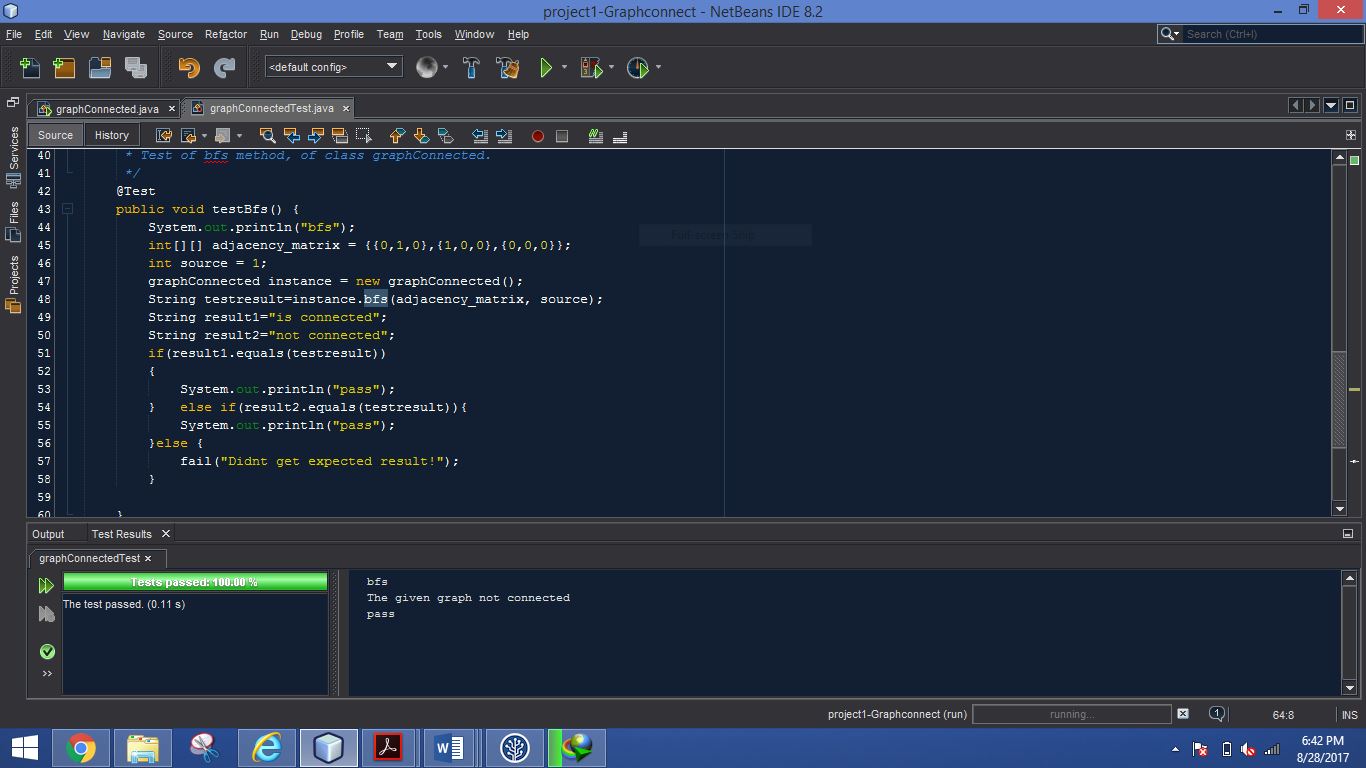
}else {

fail("Didnt get expected result!");

}

}

}



**Test pass with connect graph**

import org.junit.After;

import org.junit.AfterClass;

import org.junit.Before;

import org.junit.BeforeClass;

import org.junit.Test;

import static org.junit.Assert.\*;

/\*\*

\*

\* @author labin

\*/

public class graphConnectedTest {

public graphConnectedTest() {

}

@BeforeClass

public static void setUpClass() {

}

@AfterClass

public static void tearDownClass() {

}

@Before

public void setUp() {

}

@After

public void tearDown() {

}

/\*\*

\* Test of bfs method, of class graphConnected.

\*/

@Test

public void testBfs() {

System.out.println("bfs");

int[][] adjacency\_matrix = {{0,1,0,1},{1,0,1,1},{0,1,0,1},{1,1,1,0}};

int source = 1;

graphConnected instance = new graphConnected();

String testresult=instance.bfs(adjacency\_matrix, source);

String result1="is connected";

String result2="not connected";

if(result1.equals(testresult))

{

System.out.println("pass");

} else if(result2.equals(testresult)){

System.out.println("pass");

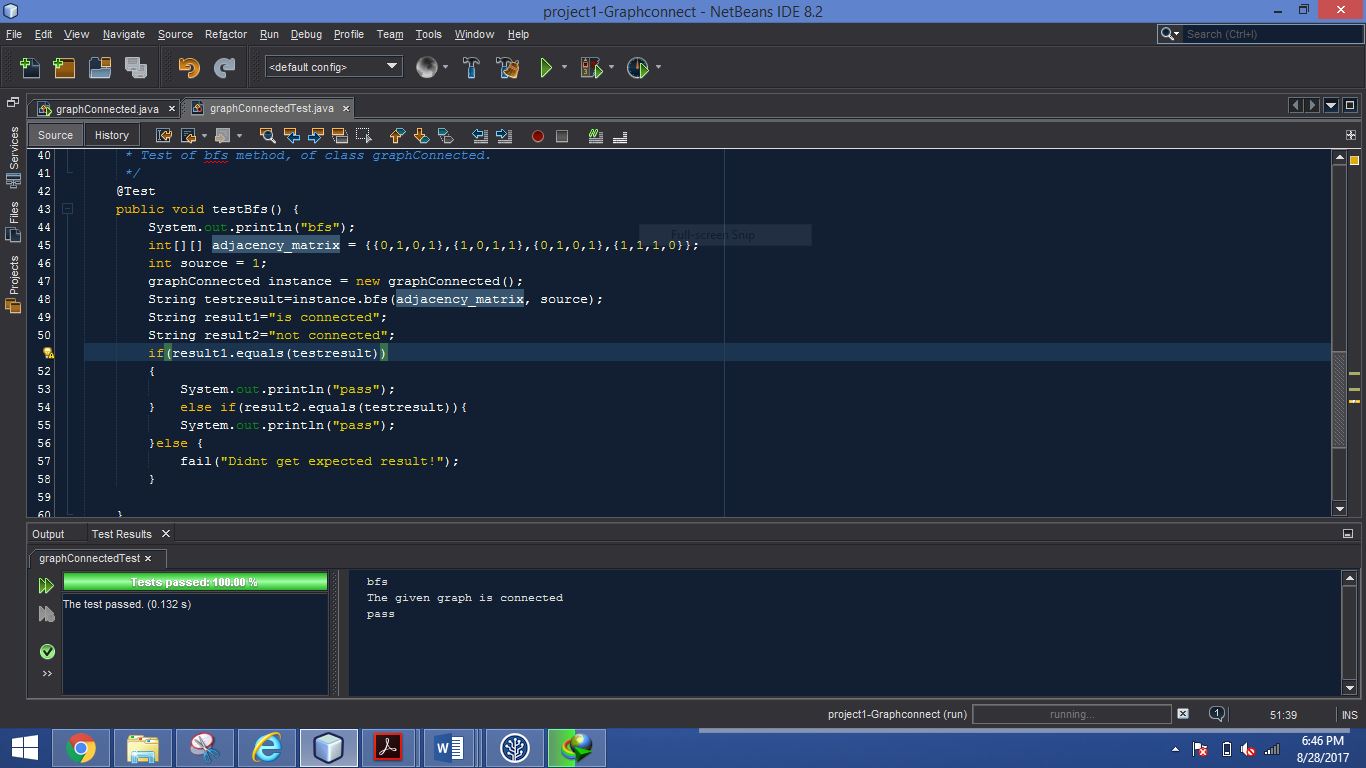
}else {

fail("Didnt get expected result!");

}

}

}

****