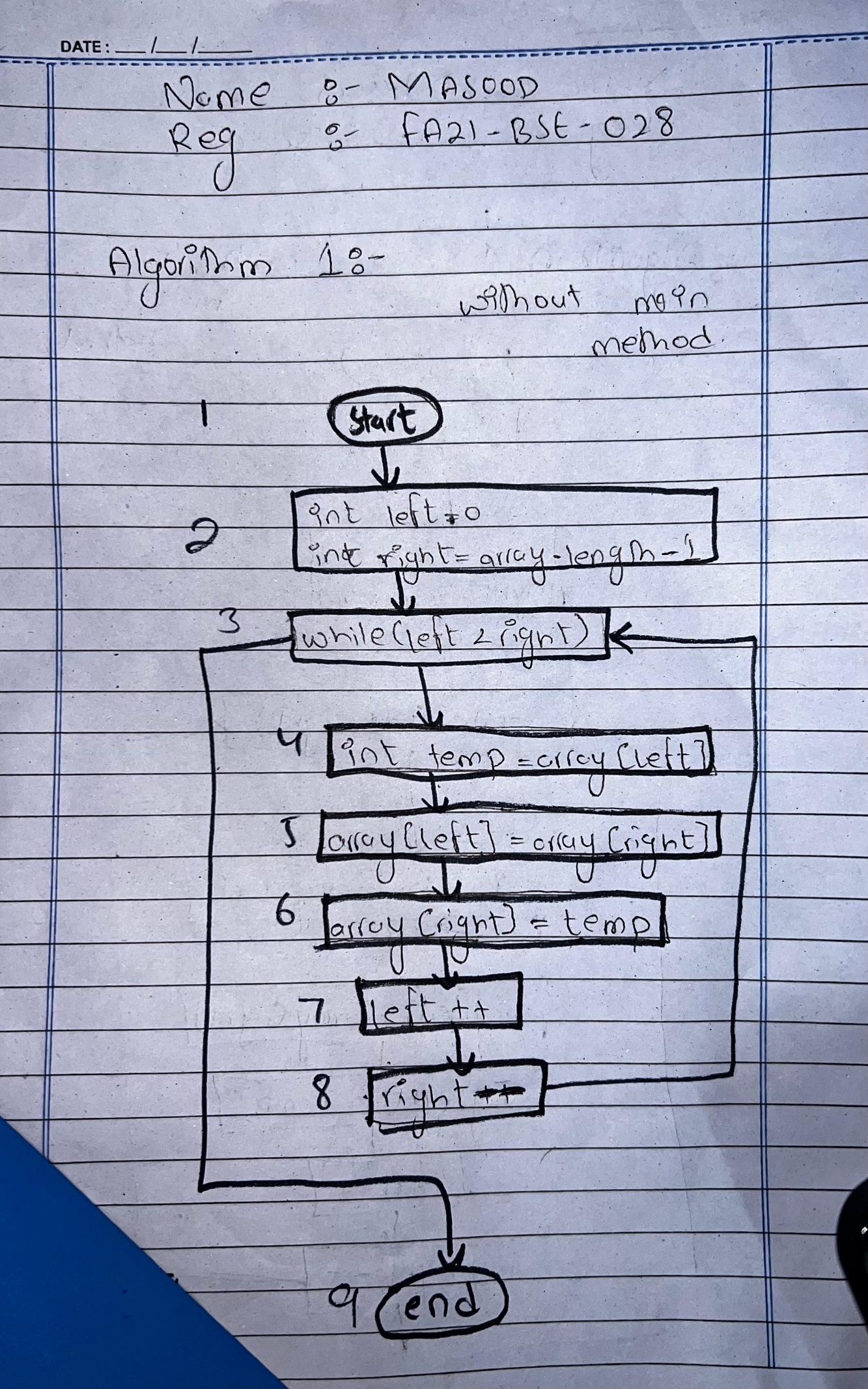
NAME : MUHAMMAD MASOOD KHAN

REGNO: FA21-BSE-028

**PATHS:**

|  |  |
| --- | --- |
| PATH | Data |
| 1,2,3,4,5,6,7,9 | 10, 20, 30, 40, 50 |
| 1,2,3,9 | {42} |

Q2) DERIVE THE TESTS AND WRITE IN TABULAR FORM

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Case Description** | **Test Case Input** | **Expected Output** | **Actual Output** | **Status** |
| TC1 | Array of odd length | 10, 20, 30, 40, 50 | 50, 40, 30, 20, 10 | 5, 4, 3, 2, 1 | Passed |
| TC2 | Array of even length | 24, 72, 93, 6 | 6, 93, 72, 24 | 6, 93, 72, 24 | Passed |
| TC3 | Empty array | {} | {} | {} | Passed |
| TC4 | Array with one element | 42 | 42 | 42 | Passed |
| TC5 | Array containing negative numbers | -17, -33, -1, -8 | -8, -1, -33, -17 | -8, -1, -33, -17 | Passed |

# **ALGORITHM1:**

I tested it in both INTELLIj IDEA and APACHE NETBEANS

package org.example;

/\*\*\*\* @author masoo\*/

public class ReverseArray {

public static void main(String[] args) {

int[] array = {1, 2, 3, 4, 5};

int[] reversedArray = reverseArray(array);

for (int num : reversedArray) {

System.out.print(num + " ");

}

}

public static int[] reverseArray(int[] array) {

int[] reversedArray = new int[array.length];

int left = 0, right = array.length - 1;

while (left < array.length) {

reversedArray[left] = array[right];

left++;

right--;

}

return reversedArray;

}

}

# **JUINT TEST CODE:**

import org.example.\*;

import org.junit.Test;

import static org.junit.Assert.assertArrayEquals;

public class Labmidtest {

@Test

public void oddlength() {

ReverseArray reverseArray = new ReverseArray();

int[] inputArray = {10, 20, 30, 40, 50};

int[] expectedArray = {50, 40, 30, 20, 10};

assertArrayEquals(expectedArray, reverseArray.reverseArray(inputArray));

}

@Test

public void evenlength() {

ReverseArray reverseArray = new ReverseArray();

int[] inputArray = {24, 72, 93, 6};

int[] expectedArray = {6, 93, 72, 24};

assertArrayEquals(expectedArray, reverseArray.reverseArray(inputArray));

}

@Test

public void testReverseArray\_EmptyArray() {

ReverseArray reverseArray = new ReverseArray();

int[] inputArray = {};

int[] expectedArray = {};

assertArrayEquals(expectedArray, reverseArray.reverseArray(inputArray));

}

@Test

public void testReverseArray\_SingleElementArray() {

ReverseArray reverseArray = new ReverseArray();

int[] inputArray = {42};

int[] expectedArray = {42};

assertArrayEquals(expectedArray, reverseArray.reverseArray(inputArray));

}

@Test

public void testReverseArray\_NegativeNumbers() {

ReverseArray reverseArray = new ReverseArray();

int[] inputArray = {-17, -33, -1, -8};

int[] expectedArray = {-8, -1, -33, -17};

assertArrayEquals(expectedArray, reverseArray.reverseArray(inputArray));

}

}

# **TEST RESULT:**

