**Software Testing**

**LAB MID**

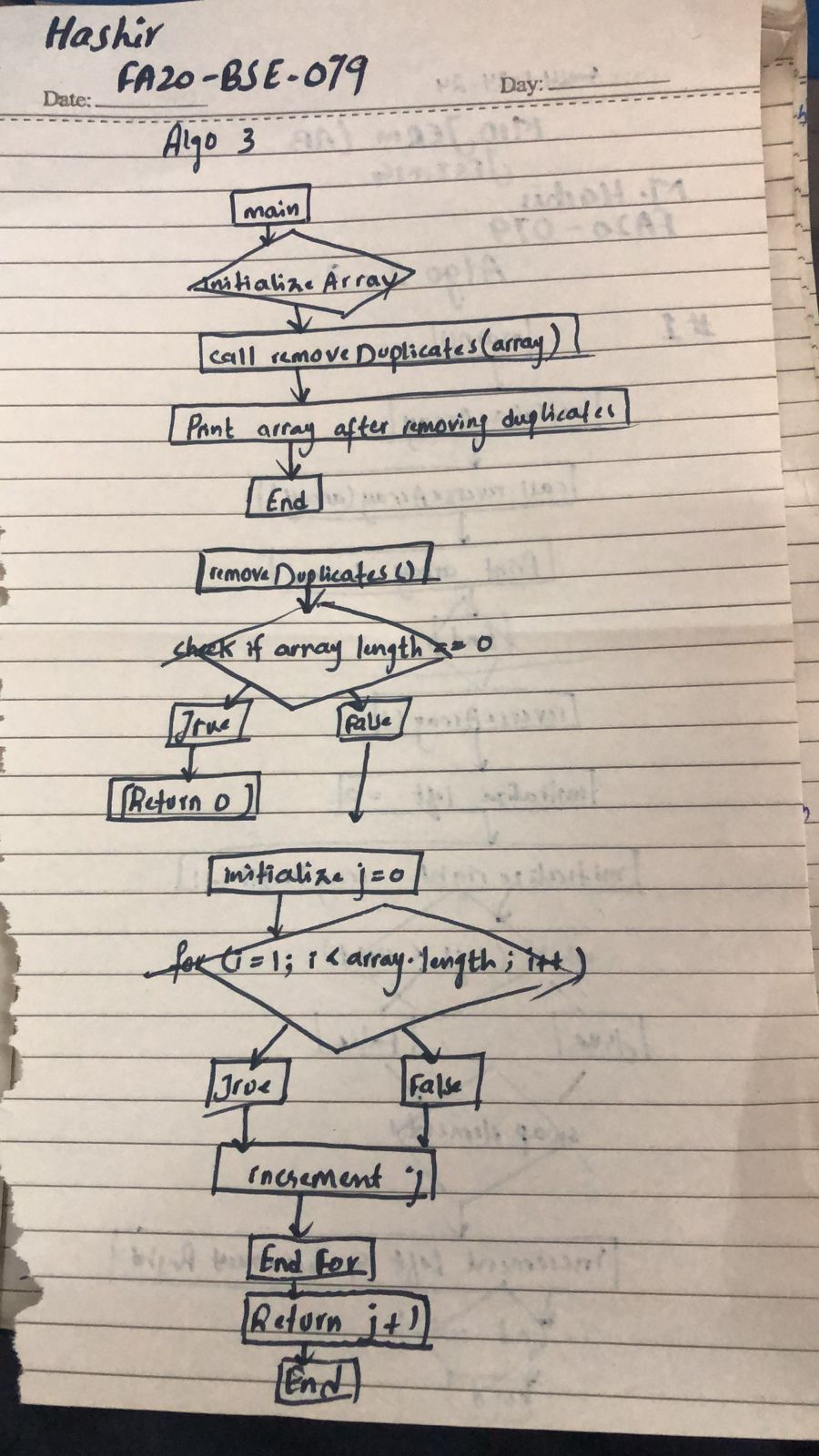
**Name:** Muhammad Hashir

**REG NO**: FA2-BSE-079

**ALGORITHM 3**

**ALGO # 3**

**Control Flow Graph**



**Test Cases**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Description** | **Input** | **Output** | **Status(Pass/fail)** |
| Tc\_01 | Remove duplicates from an empty array | {} | {} | Pass |
| Tc\_02 | Remove duplication from a single element array | {1} | {1} | Pass |
| Tc\_03 | Remove duplication from an array with all duplicates | {1,2,2,3} | {1,2,2,3} | Pass |
| Tc\_04 | Remove duplicates from an array with some duplicates | {1,2,2,3,4,4,5} | {1,2,2,3,4,4,5} | Pass |

**Junit Test:**

qimport org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.assertArrayEquals;

public class RemoveDuplicatesTest {

@Test

public void testRemoveDuplicatesEmptyArray() {

int[] input = {};

int expectedLength = 0;

int newLength = RemoveDuplicates.removeDuplicates(input);

assertArrayEquals(input, Arrays.copyOf(input, newLength));

}

@Test

public void testRemoveDuplicatesSingleElementArray() {

int[] input = {1};

int expectedLength = 1;

int newLength = RemoveDuplicates.removeDuplicates(input);

assertArrayEquals(input, Arrays.copyOf(input, newLength));

}

@Test

public void testRemoveDuplicatesAllDuplicates() {

int[] input = {1, 1, 1, 1};

int expectedLength = 1;

int[] expectedArray = {1};

int newLength = RemoveDuplicates.removeDuplicates(input);

assertArrayEquals(expectedArray, Arrays.copyOf(input, newLength));

}

@Test

public void testRemoveDuplicatesSomeDuplicates() {

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

int expectedLength = 5;

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

int newLength = RemoveDuplicates.removeDuplicates(input);

assertArrayEquals(expectedArray, Arrays.copyOf(input, newLength));

}

}