Software testing Mid Term Lab

Name: Uzair Ali

Reg No: SP21-BSE-025

Teacher: Sir, Mukhtiar Zamin

Algorithm

Reverse an Array: Swaps elements from the start and end moving towards the center to reverse the array.

Control Flow Graph

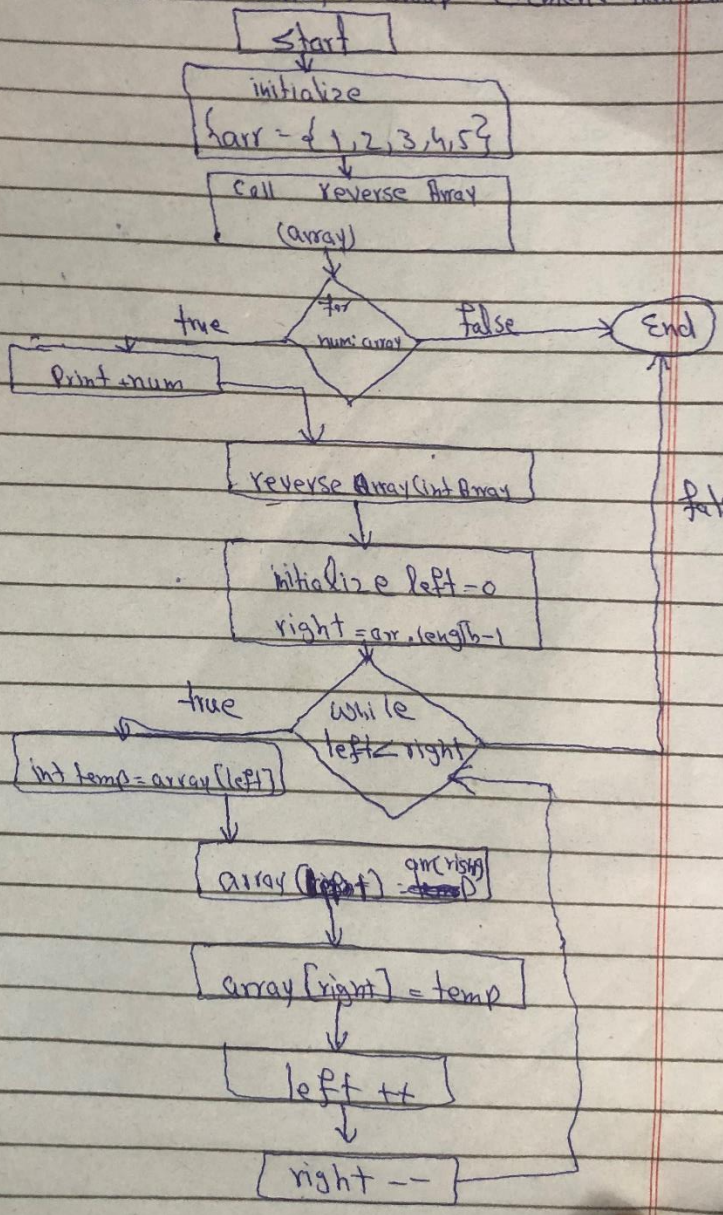
Day: Date:

Name: UZAIR ALI

Reg No: SP21-BSE-025

Algorithm No 1:

Reverse An Array, Swap Element from start and end



Test Cases:

Test Case Id	Description	Input	Expected Output	Actual Outcome	Status
1	Test with a simple	{1, 2, 3, 4, 5}	5 4 3 2 1	5 4 3 2 1	Pass

	array of 5 elements.				
2	**Test with a simple array of 5 elements with larger values.**	**{10, 20, 30, 40, 50}**	**50 40 30 20 10**	**50 40 30 20 10**	**Pass**
3	**Test with an array of 4 elements.**	**{1, 2, 3, 4}**	**4 3 2 1**	**4 3 2 1**	**Pass**
4	**Test with an array of 5 elements with the same value.**	**{5, 5, 5, 5, 5}**	**5 5 5 5 5**	**5 5 5 5 5**	**Pass**
5	**(no output) Test with an empty array**	**{}**	**Invalid (Empty Array)**	**Invalid (Empty Array)**	**Pass**

Junit: Algorithm

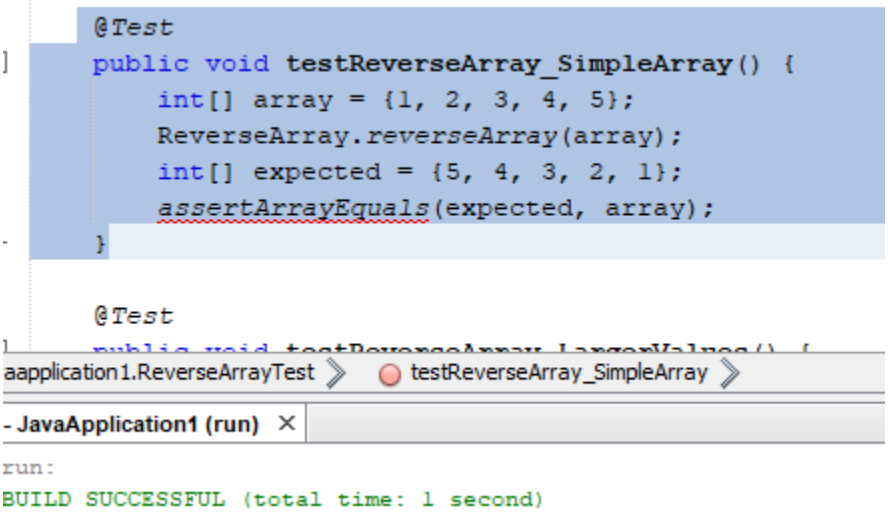
```
package javaapplication1;

import org.junit.Test;
```

```
import static org.junit.Assert.assertArrayEquals;

public class ReverseArrayTest {

    @Test
    public void testReverseArray_SimpleArray() {
        int[] array = {1, 2, 3, 4, 5};
        ReverseArray.reverseArray(array);
        int[] expected = {5, 4, 3, 2, 1};
        assertArrayEquals(expected, array);
    }

    @Test
    public void testReverseArray_LargerValues() {
        int[] array = {10, 20, 30, 40, 50};
        ReverseArray.reverseArray(array);
        int[] expected = {50, 40, 30, 20, 10};
        assertArrayEquals(expected, array);
    }

    @Test
    public void testReverseArray_ArrayOfFourElements() {
        int[] array = {1, 2, 3, 4};
        ReverseArray.reverseArray(array);
```

```
    int[] expected = {4, 3, 2, 1};
    assertArrayEquals(expected, array);
}

@Test
public void testReverseArray_ArrayOfSameValues() {
    int[] array = {5, 5, 5, 5, 5};
    ReverseArray.reverseArray(array);
    int[] expected = {5, 5, 5, 5, 5};
    assertArrayEquals(expected, array);
}

@Test
public void testReverseArray_EmptyArray() {
    int[] array = {};
    ReverseArray.reverseArray(array);
    int[] expected = {};
    assertArrayEquals(expected, array);
}

@Test
public void testReverseArray_ArrayOfOneElement() {
    int[] array = {1};
    ReverseArray.reverseArray(array);
```

```
    int[] expected = {1};
    assertArrayEquals(expected, array);
  }

  @Test
  public void testReverseArray_ArrayOfTwoElements() {
    int[] array = {1, 2};
    ReverseArray.reverseArray(array);
    int[] expected = {2, 1};
    assertArrayEquals(expected, array);
  }
}
```

Unit Testing:



```java
    @Test
    public void testReverseArray_LargerValues() {
        int[] array = {10, 20, 30, 40, 50};
        ReverseArray.reverseArray(array);
        int[] expected = {50, 40, 30, 20, 10};
        assertArrayEquals(expected, array);
    }

    @Test
    public void testReverseArray_ArrayOfFourElements
```

vaapplication1.ReverseArrayTest > testReverseArray_LargerValues >

- JavaApplication1 (run) ×

```
run:
BUILD SUCCESSFUL (total time: 1 second)
```

```java
@Test
public void testReverseArray_ArrayOfSameValues() {
    int[] array = {5, 5, 5, 5, 5};
    ReverseArray.reverseArray(array);
    int[] expected = {5, 5, 5, 5, 5};
    assertArrayEquals(expected, array);
}

@Test
public void testReverseArray_EmptyArray() {
    int[] array = {};
    ReverseArray.reverseArray(array);
```

vaapplication1.ReverseArrayTest > testReverseArray_ArrayOfSameValues >

- JavaApplication1 (run) ×

```
run:
BUILD SUCCESSFUL (total time: 1 second)
```

```java
@Test
public void testReverseArray_ArrayOfSameValues() {
    int[] array = {5, 5, 5, 5, 5};
    ReverseArray.reverseArray(array);
    int[] expected = {5, 5, 5, 5, 5};
    assertArrayEquals(expected, array);
}

@Test
public void testReverseArray_EmptyArray() {
    int[] array = {};
    ReverseArray.reverseArray(array);
```

aapplication1.ReverseArrayTest > testReverseArray_ArrayOfFourElements >

- JavaApplication1 (run) ×

```
run:
BUILD SUCCESSFUL (total time: 1 second)
```

```java
@Test
public void testReverseArray_ArrayOfFourElements() {
    int[] array = {1, 2, 3, 4};
    ReverseArray.reverseArray(array);
    int[] expected = {4, 3, 2, 1};
    assertArrayEquals(expected, array);
}

@Test
public void testReverseArray_ArrayOfSameValues() {
    int[] array = {5, 5, 5, 5, 5};
    ReverseArray.reverseArray(array);
    int[] expected = {5, 5, 5, 5, 5};
    assertArrayEquals(expected, array);
```

avaapplication1.ReverseArrayTest › testReverseArray_ArrayOfFourElements

t - JavaApplication1 (run)

```
run:
BUILD SUCCESSFUL (total time: 1 second)
```

According to test cases and Junit Testing the Expected outcome is equal to the actual outcome