COMSATS UNIVERSITY ISLAMABAD, ABBOTTABAD

Software Testing

Lab Mid

IntelliJ IDE

Submitted by:

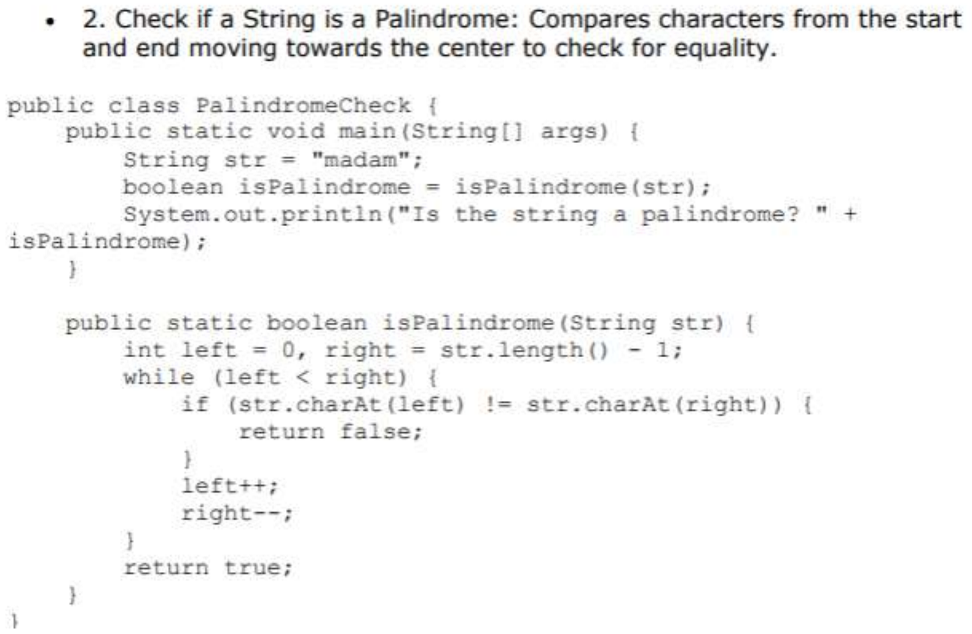
Arfah Ali FA21-BSE-080

Submitted to:

Sir Mukhtiar Zamin

Algorithm No. 02

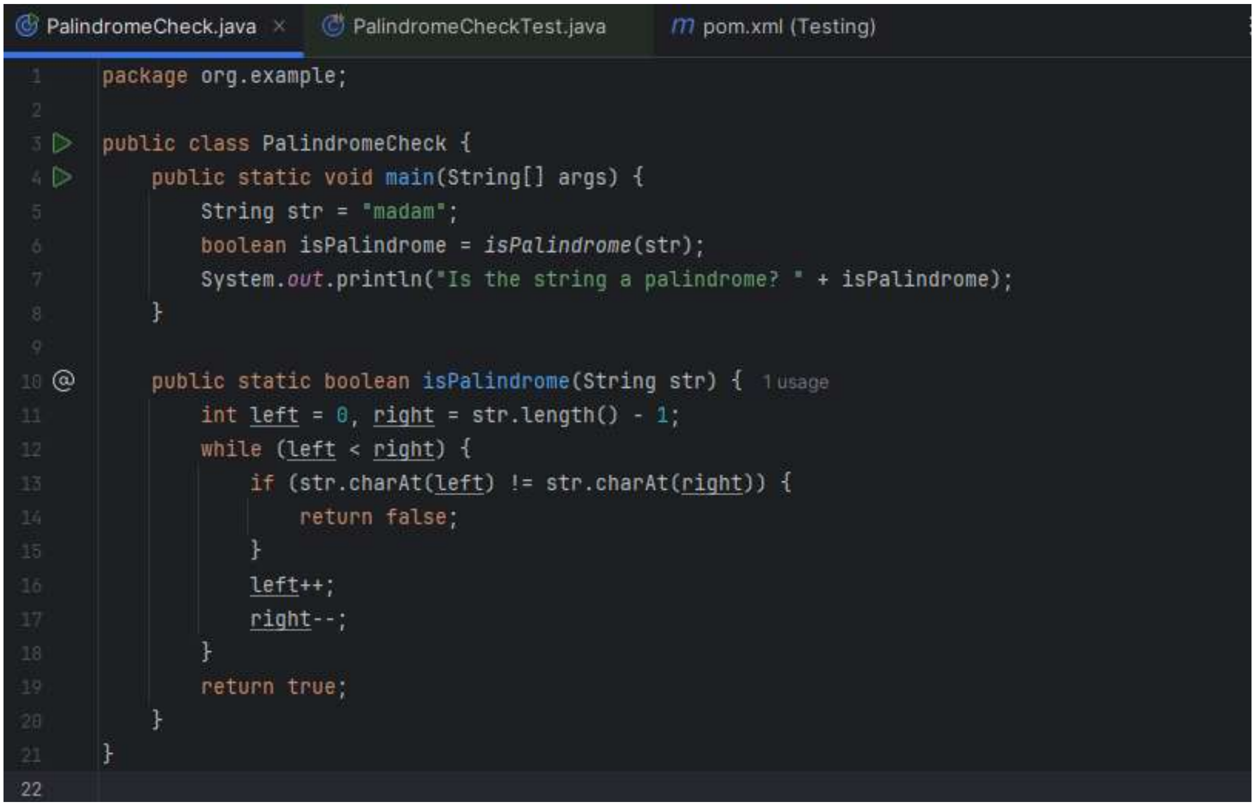
I have been assigned with the task of solving an Algorithm 2 which was an algo to check whether the given string is a palindrome or not. I have sketched a CFG, paths regarding that CFG and then draw some testcase that pass through those paths. All of them were passed through the given algorithm. I have done this task using IntelliJ IDE.



- 2. Check if a String is a Palindrome: Compares characters from the start and end moving towards the center to check for equality.

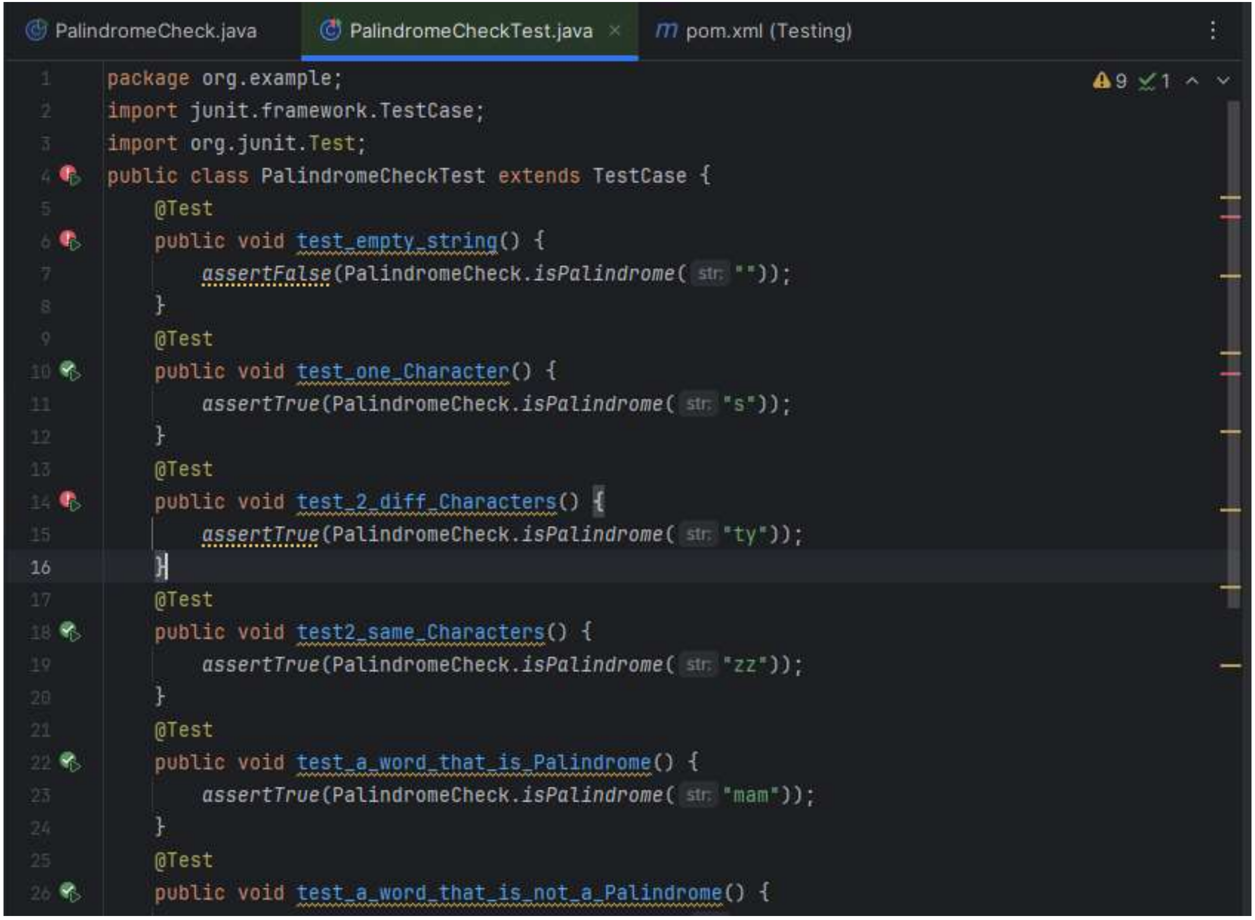
```
public class PalindromeCheck {
    public static void main(String[] args) {
        String str = "madam";
        boolean isPalindrome = isPalindrome(str);
        System.out.println("Is the string a palindrome? " +
isPalindrome);
    }

    public static boolean isPalindrome(String str) {
        int left = 0, right = str.length() - 1;
        while (left < right) {
            if (str.charAt(left) != str.charAt(right)) {
                return false;
            }
            left++;
            right--;
        }
        return true;
    }
}
```

PalindromeCheck.java

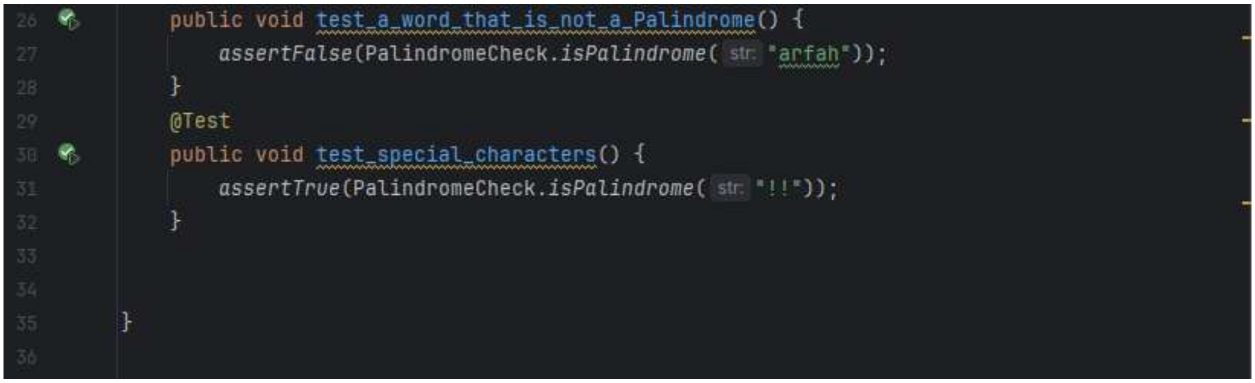


```java
package org.example;

public class PalindromeCheck {
    public static void main(String[] args) {
        String str = "madam";
        boolean isPalindrome = isPalindrome(str);
        System.out.println("Is the string a palindrome? " + isPalindrome);
    }

    public static boolean isPalindrome(String str) {
        int left = 0, right = str.length() - 1;
        while (left < right) {
            if (str.charAt(left) != str.charAt(right)) {
                return false;
            }
            left++;
            right--;
        }
        return true;
    }
}
```

PalindromeCheckTest.java

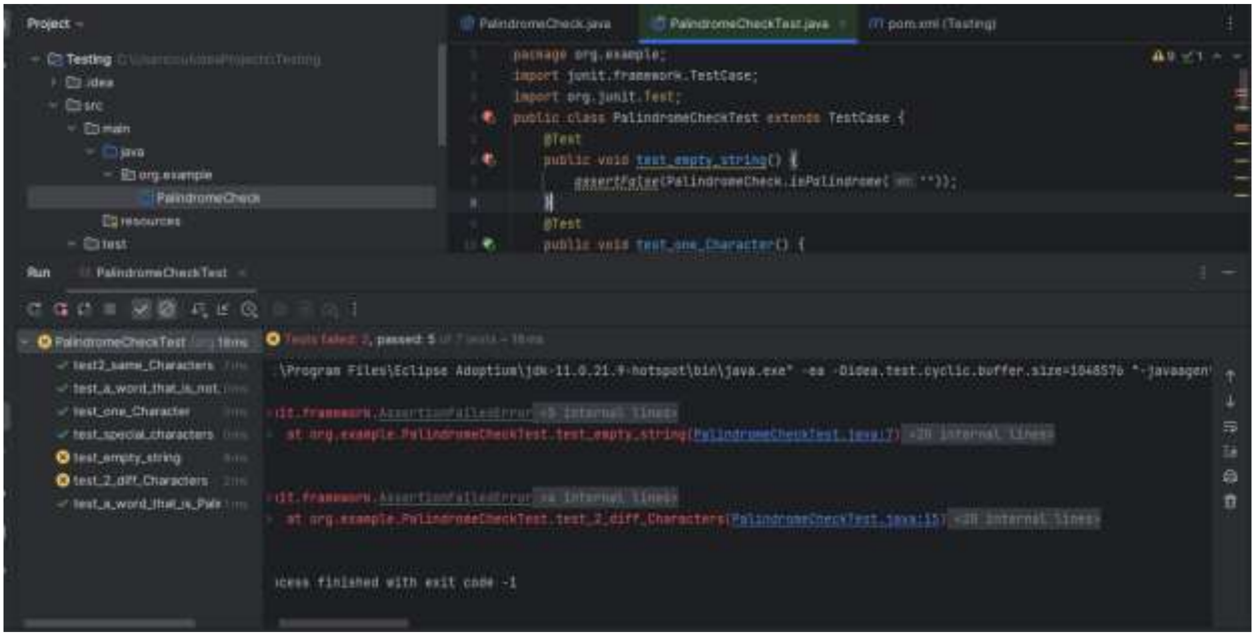


```
package org.example;
import junit.framework.TestCase;
import org.junit.Test;
public class PalindromeCheckTest extends TestCase {
    @Test
    public void test_empty_string() {
        assertFalse(PalindromeCheck.isPalindrome(""));
    }
    @Test
    public void test_one_Character() {
        assertTrue(PalindromeCheck.isPalindrome("s"));
    }
    @Test
    public void test_2_diff_Characters() {
        assertTrue(PalindromeCheck.isPalindrome("ty"));
    }
    @Test
    public void test2_same_Characters() {
        assertTrue(PalindromeCheck.isPalindrome("zz"));
    }
    @Test
    public void test_a_word_that_is_Palindrome() {
        assertTrue(PalindromeCheck.isPalindrome("mam"));
    }
    @Test
    public void test_a_word_that_is_not_a_Palindrome() {
```



```
    public void test_a_word_that_is_not_a_Palindrome() {
        assertFalse(PalindromeCheck.isPalindrome("arfah"));
    }
    @Test
    public void test_special_characters() {
        assertTrue(PalindromeCheck.isPalindrome("!!"));
    }


}
```

Tested Results:

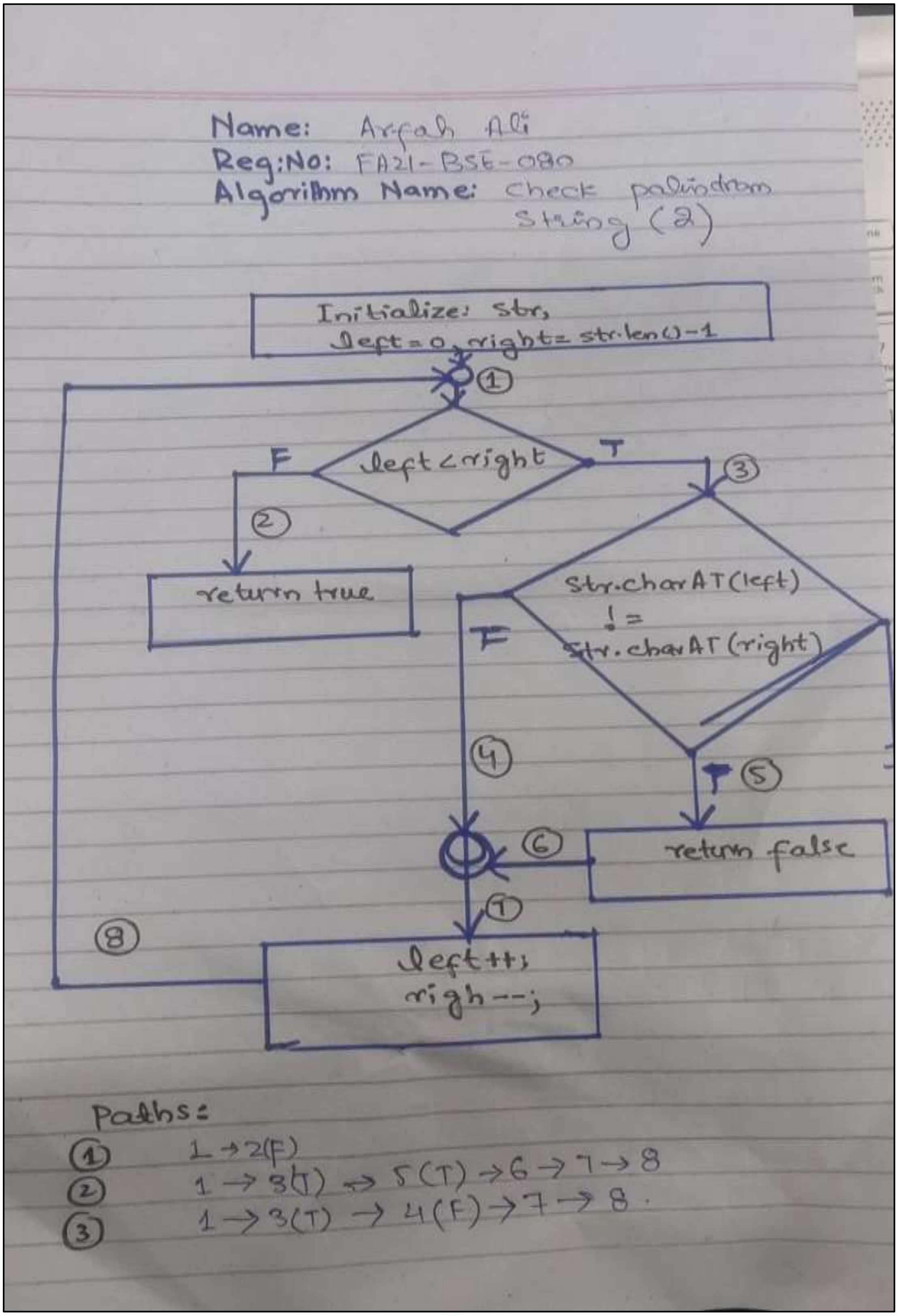


```java
package org.example;
import junit.framework.TestCase;
import org.junit.Test;
public class PalindromeCheckTest extends TestCase {
    @Test
    public void test_empty_string() {
        assertFalse(PalindromeCheck.isPalindrome( str: ""));
    }
    @Test
    public void test_one_Character() {
```

```
PalindromeCheckTest 18ms
  ✓ test2_same_Characters 7ms
  ✓ test_a_word_that_is_not_
  ✓ test_one_Character
  ✓ test_special_characters
  ☒ test_empty_string 8ms
  ☒ test_2_diff_Characters 2ms
  ✓ test_a_word_that_is_Pali 1ms
```

```
Tests failed: 2, passed: 5 of 7 tests – 18ms

\Program Files\Eclipse Adoptium\jdk-11.0.21.9-hotspot\bin\java.exe" -ea -Didea.test.cyclic.buffer.size=1048576 "-javaagen

it.framework.AssertionFailedError <3 internal lines>
  at org.example.PalindromeCheckTest.test_empty_string(PalindromeCheckTest.java:7) <20 internal lines>


it.framework.AssertionFailedError <4 internal lines>
  at org.example.PalindromeCheckTest.test_2_diff_Characters(PalindromeCheckTest.java:15) <20 internal lines>


cess finished with exit code -1
```

CFG:



Name: Arfah Ali
Reg:No: FA21-BSE-080
Algorithm Name: check palindrom String (2)

Paths:

① 1 → 2(F)
② 1 → 3(T) → 5(T) → 6 → 7 → 8
③ 1 → 3(T) → 4(F) → 7 → 8.

Paths:

1. **Path 1:** 1 → 2 (F)
2. **Path 2:** 1 → 3 (T) → 5 (T) → 6 → 7 → 8
3. **Path 3:** 1 → 3 (T) → 4 (F) → 7 → 8

Test Cases:

I gave some wrong inputs so that program passes through the False paths as well. One with one string, One with multiple but wrong so that it passes the condition inside the loop, and one with more than one correct order of characters. I have checked special characters order and an empty string as well. Check the implementation in the provided code.

Test Case ID	**Description**	**Input Data**	**Expected outcome**	**Actual outcome**	**Status/Verdict**
TC1	Keep it an empty string, which is a palindrome.	""	It is a palindrome	It is not a palindrome	Fail
TC2	Enter one character which is also a palindrome.	"s"	It is a palindrome	It is a palindrome	Pass
TC3	Enter 2 different characters which do not make palindrome.	"ty"	It is not a palindrome	It is a palindrome	Fail
TC4	Enter 2 same characters which make a palindrome string.	"zz"	It is a palindrome	It is a palindrome	Pass
TC5	Enter a word that is a palindrome.	"mam"	It is a palindrome	It is a palindrome	Pass
TC6	Enter a complete word that is not a palindrome.	"arfah"	It is not a palindrome	It is not a palindrome	pass
TC7	Enter special characters	“!!”	It is a palindrome	It is a palindrome	pass

THE END