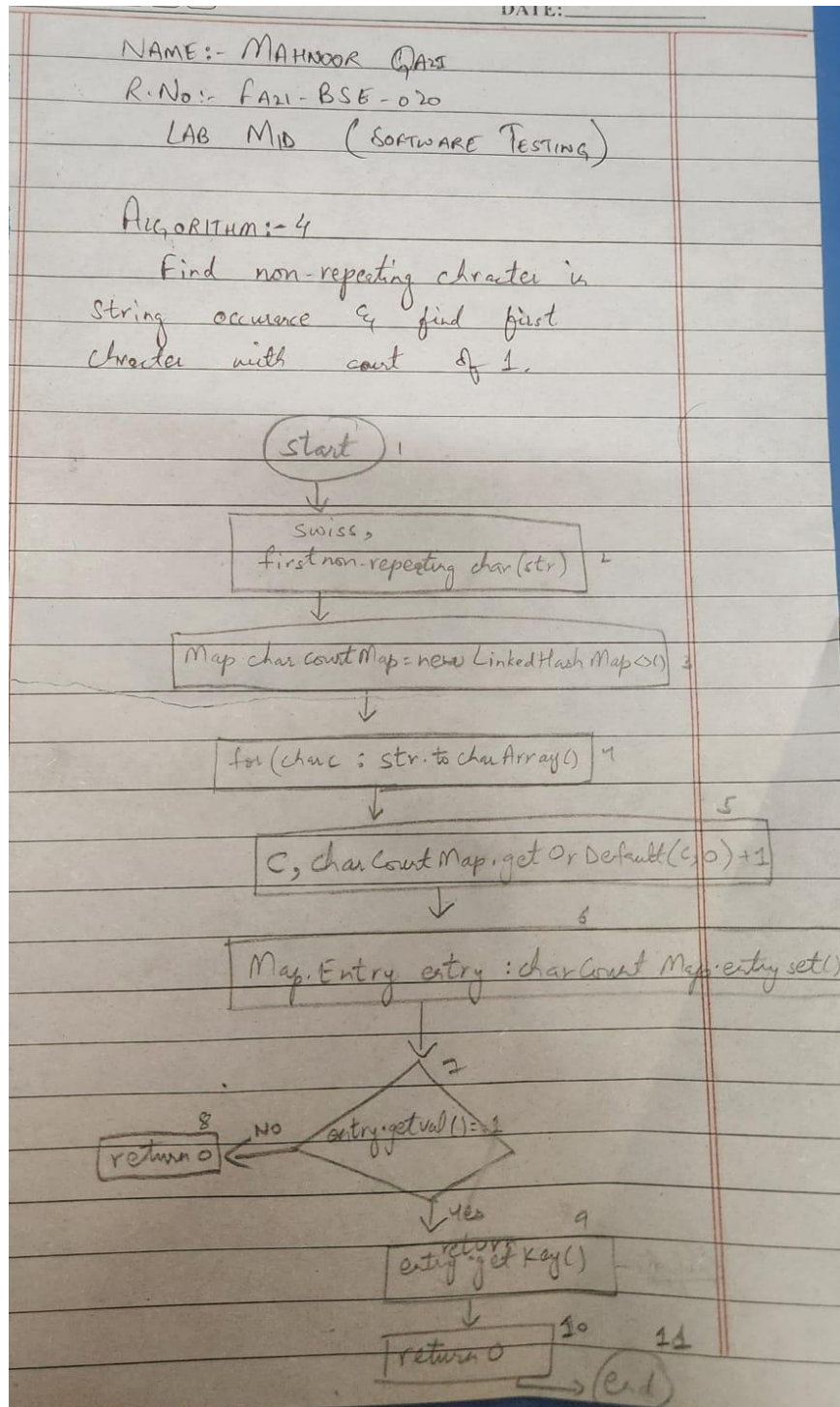


NAME: MAHNOOR QAZI**R.NO: FA21-BSE-020****SUBJECT: SOFTWARE TESTING****LAB.MID**

ST-LABMID

PATH:

1-2-3-4-5-6-7-8

PATH 2:

1-2-3-4-5-6-7-9-10-11

TEST CASES FOR ALGO ARE GIVEN BELOW:

Test Case ID	Description	Input Data	Expected Outcome	Actual Outcome	Status
TC-1	String with repeating character	swiss	s	s	pass
TC-2	String with non-repeating character	swiss	w	w	pass
TC-3	String with no character	empty	empty	empty	pass
TC-4	First unique character in string	swiss	w	w	pass
TC-5	First unique character in string	swiss	w	i	fail
TC-6	String with repeating character	swiss	s	i	fail
TC-7	String with no character	empty	empty	s	fail

Junit:

ST-LABMID

```

1  import java.util.LinkedHashMap;
2  import java.util.Map;
3
4  public class FirstNonRepeatingChar {
5      public static void main(String[] args) {
6          String str = "swiss";
7          char result = firstNonRepeatingChar(str);
8          System.out.println("First non-repeating character is: " + result);
9      }
10
11     public static char firstNonRepeatingChar(String str) {
12         Map<Character, Integer> charCountMap = new LinkedHashMap<>();
13         for (char c : str.toCharArray()) {
14             charCountMap.put(key:c, charCountMap.getOrDefault(key:c, defaultValue: 0) + 1);
15         }
16         for (Map.Entry<Character, Integer> entry : charCountMap.entrySet()) {
17             if (entry.getValue() == 1) {
18                 return entry.getKey();
19             }
20         }
21         return '\0'; // Indicates no non-repeating character found
22     }
23 }

```

TEST:

```
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

public class FirstNonRepeatingCharTest {

    @Test
    public void testFirstNonRepeatingChar_RepeatingCharacter() {
        String input = "swiss";
        char expected = 's';
        char actual = FirstNonRepeatingChar.firstNonRepeatingChar(str:input);
        assertEquals(expected, actual);
    }

    @Test
    public void testFirstNonRepeatingChar_NonRepeatingCharacter() {
        String input = "swiss";
        char expected = 'w';
        char actual = FirstNonRepeatingChar.firstNonRepeatingChar(str:input);
        assertEquals(expected, actual);
    }

    @Test
    public void testFirstNonRepeatingChar_EmptyString() {
        String input = "";
        char expected = '\0';
        char actual = FirstNonRepeatingChar.firstNonRepeatingChar(str:input);
        assertEquals(expected, actual);
    }
}
```

```

@Test
public void testFirstNonRepeatingChar_FirstUniqueCharacter() {
    String input = "swiss";
    char expected = 'w';
    char actual = FirstNonRepeatingChar.firstNonRepeatingChar(str::input);
    assertEquals(expected, actual);
}

@Test
public void testFirstNonRepeatingChar_FirstUniqueCharacter_Fail() {
    String input = "swiss";
    char unexpected = 'i'; // This should fail
    char actual = FirstNonRepeatingChar.firstNonRepeatingChar(str::input);
    assertNotEquals(unexpected, actual);
}

@Test
public void testFirstNonRepeatingChar_RepeatingCharacter_Fail() {
    String input = "swiss";
    char unexpected = 'i'; // This should fail
    char actual = FirstNonRepeatingChar.firstNonRepeatingChar(str::input);
    assertNotEquals(unexpected, actual);
}

@Test
public void testFirstNonRepeatingChar_EmptyString_Fail() {
    String input = "";
    char unexpected = 's'; // This should fail
    char actual = FirstNonRepeatingChar.firstNonRepeatingChar(str::input);

```

```

6      @Test
7      public void testFirstNonRepeatingChar_RepeatingCharacter_Fail() {
8          String input = "swiss";
9          char unexpected = 'i'; // This should fail
10         char actual = FirstNonRepeatingChar.firstNonRepeatingChar(str::input);
11         assertNotEquals(unexpected, actual);
12     }
13
14     @Test
15     public void testFirstNonRepeatingChar_EmptyString_Fail() {
16         String input = "";
17         char unexpected = 's'; // This should fail
18         char actual = FirstNonRepeatingChar.firstNonRepeatingChar(str::input);
19         assertNotEquals(unexpected, actual);
20     }
21 }
22
23

```

OUTPUT:

ST-LABMID

