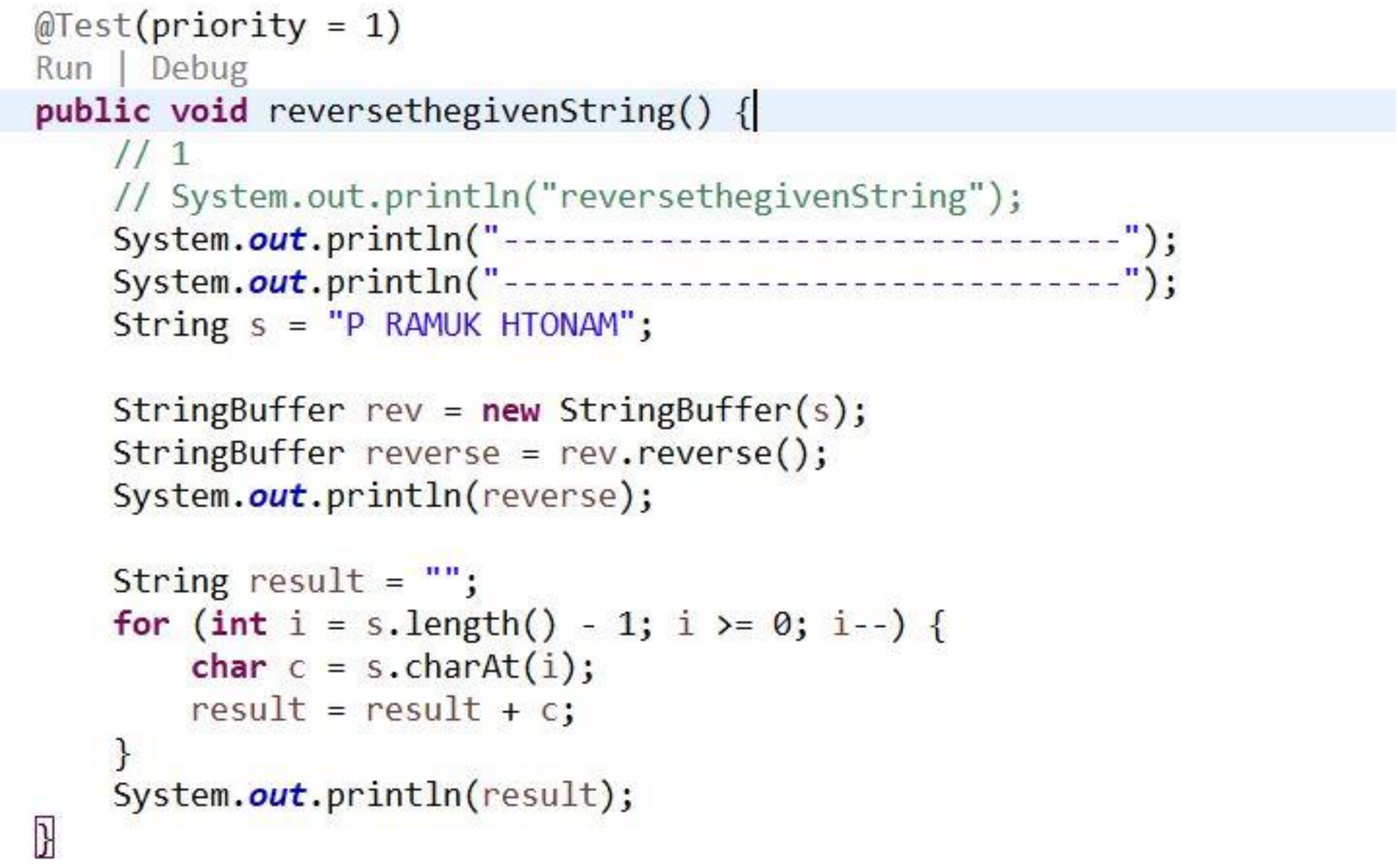
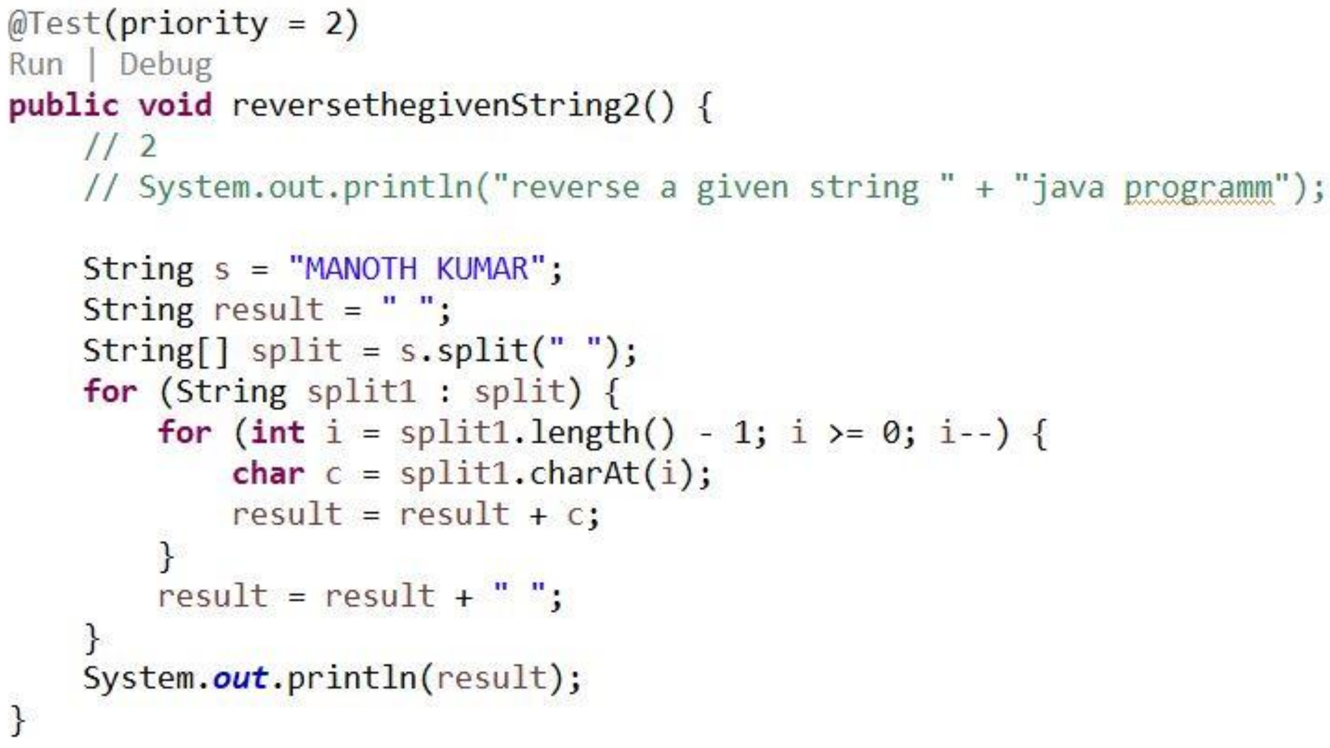
JAVA LOGICS ~ MK

1. REVERSE THE STRING ENTIRE



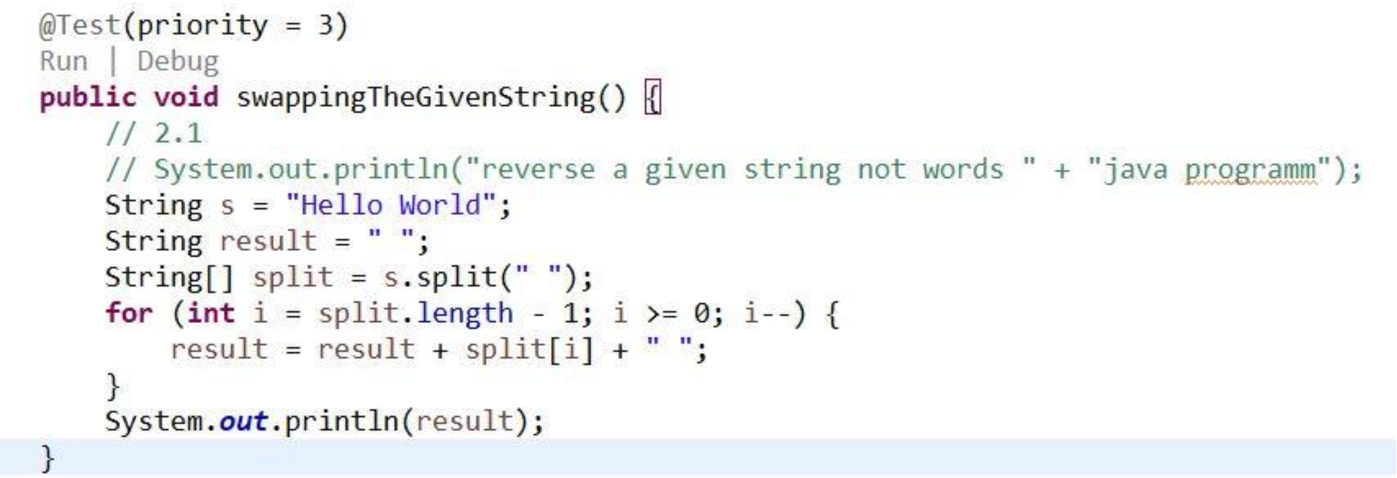
```java
@Test(priority = 1)
Run | Debug
public void reversethegivenString() {
	// 1
	// System.out.println("reversethegivenString");
	System.out.println("--------------------------------");
	System.out.println("--------------------------------");
	String s = "P RAMUK HTONAM";

	StringBuffer rev = new StringBuffer(s);
	StringBuffer reverse = rev.reverse();
	System.out.println(reverse);

	String result = "";
	for (int i = s.length() - 1; i >= 0; i--) {
		char c = s.charAt(i);
		result = result + c;
	}
	System.out.println(result);
}
```

2. REVERSE THE STRING WORDS



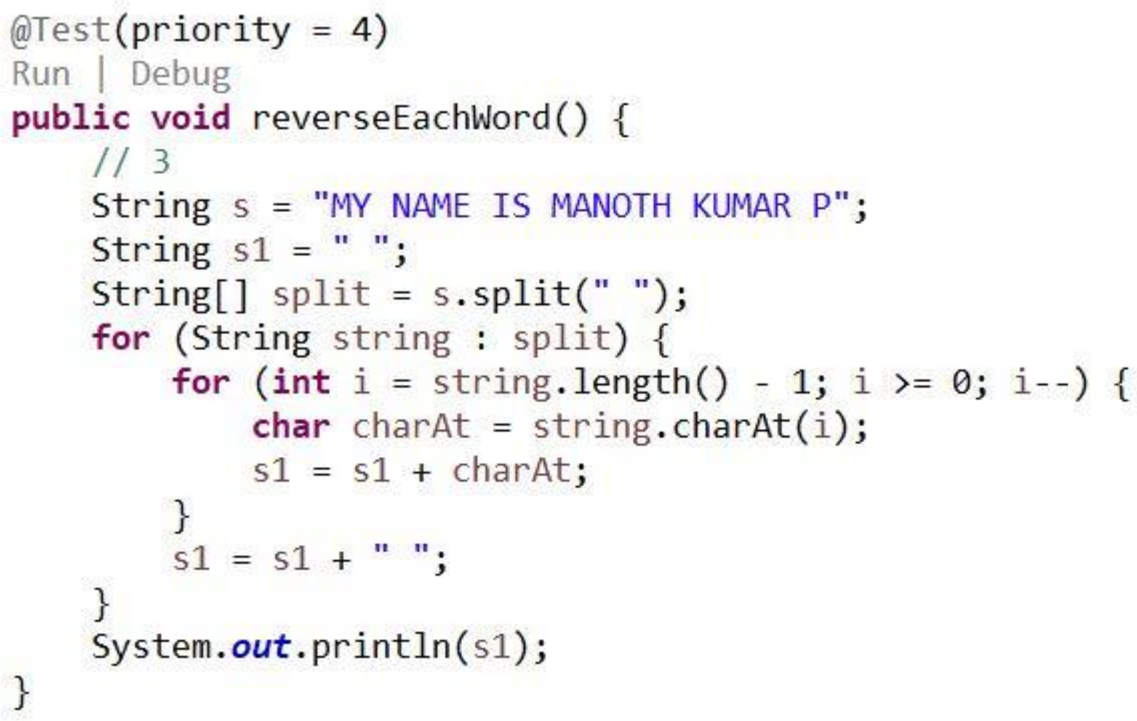
```java
@Test(priority = 2)
Run | Debug
public void reversethegivenString2() {
	// 2
	// System.out.println("reverse a given string " + "java programm");

	String s = "MANOTH KUMAR";
	String result = " ";
	String[] split = s.split(" ");
	for (String split1 : split) {
		for (int i = split1.length() - 1; i >= 0; i--) {
			char c = split1.charAt(i);
			result = result + c;
		}
		result = result + " ";
	}
	System.out.println(result);
}
```

3. SWAPPING THE STRING



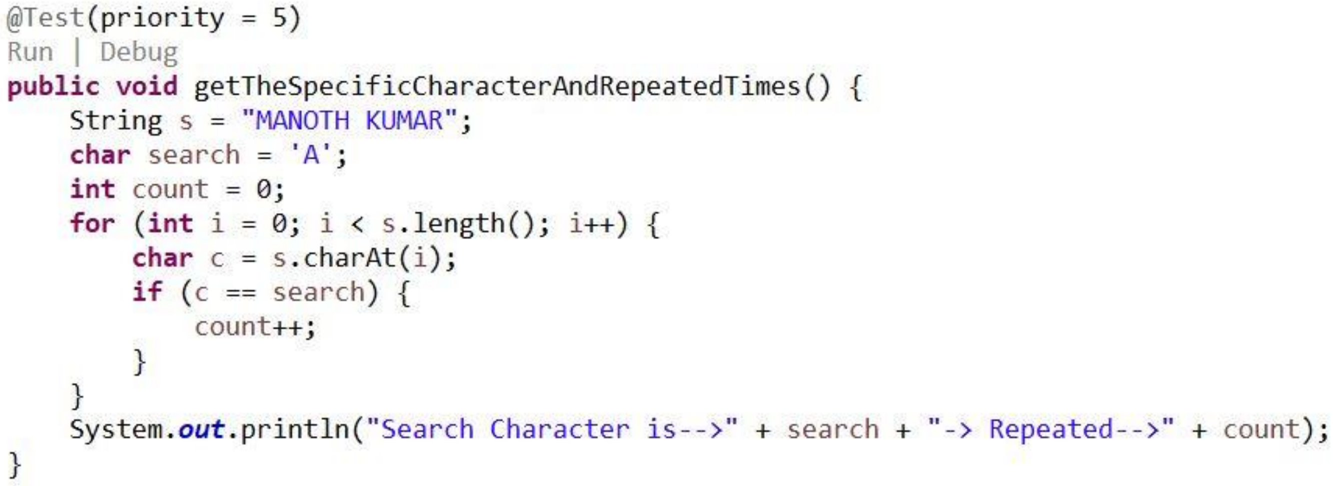
```
@Test(priority = 3)
Run | Debug
public void swappingTheGivenString() {
    // 2.1
    // System.out.println("reverse a given string not words " + "java programm");
    String s = "Hello World";
    String result = " ";
    String[] split = s.split(" ");
    for (int i = split.length - 1; i >= 0; i--) {
        result = result + split[i] + " ";
    }
    System.out.println(result);
}
```

4. REVERSE THE EACH WORD STRING



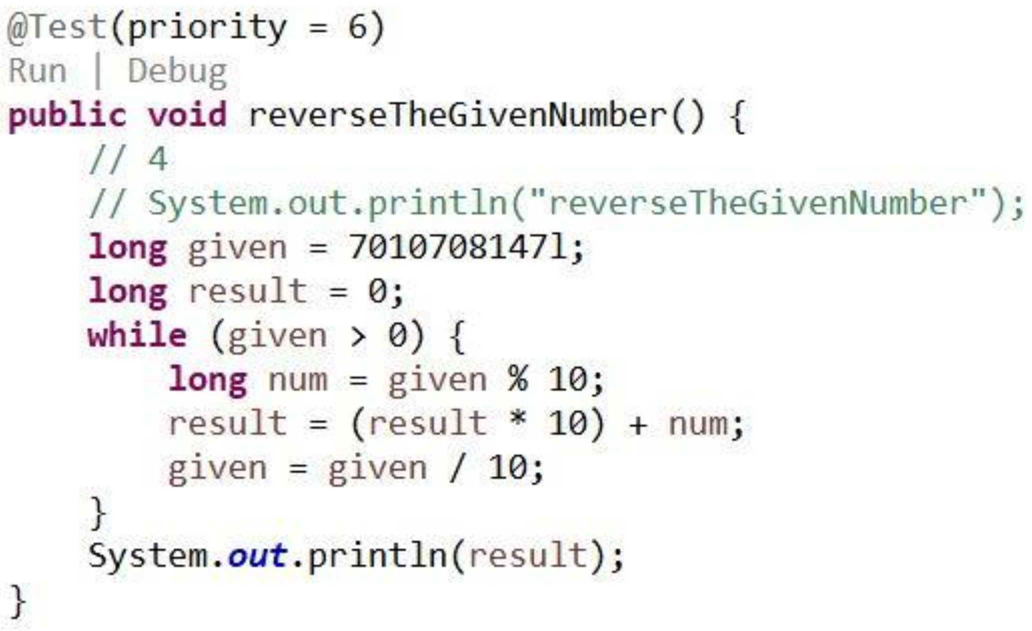
```
@Test(priority = 4)
Run | Debug
public void reverseEachWord() {
    // 3
    String s = "MY NAME IS MANOTH KUMAR P";
    String s1 = " ";
    String[] split = s.split(" ");
    for (String string : split) {
        for (int i = string.length() - 1; i >= 0; i--) {
            char charAt = string.charAt(i);
            s1 = s1 + charAt;
        }
        s1 = s1 + " ";
    }
    System.out.println(s1);
}
```

5. CHAR REPEATED IN STRING



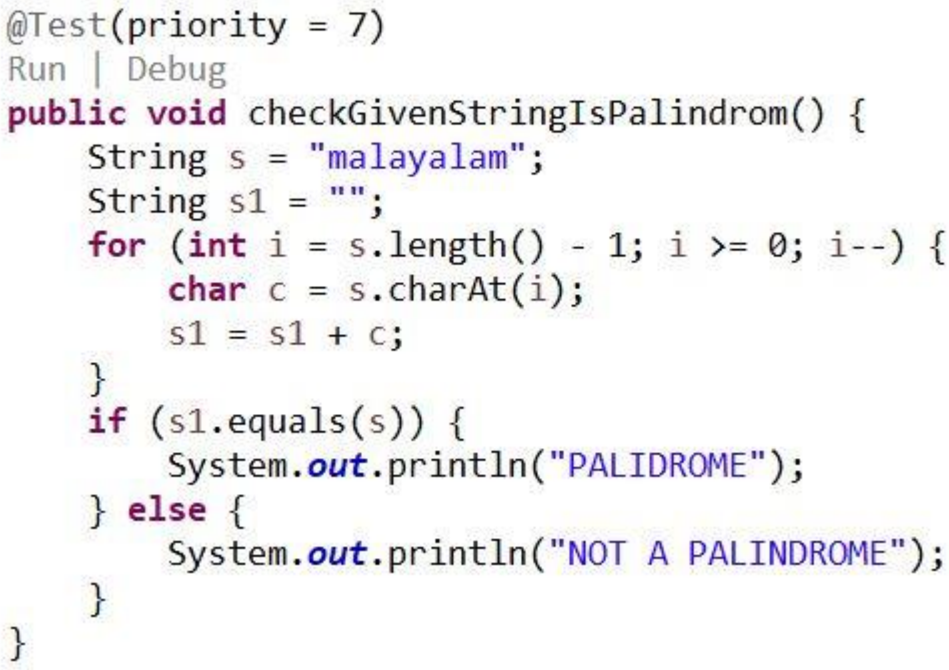
```
@Test(priority = 5)
Run | Debug
public void getTheSpecificCharacterAndRepeatedTimes() {
    String s = "MANOTH KUMAR";
    char search = 'A';
    int count = 0;
    for (int i = 0; i < s.length(); i++) {
        char c = s.charAt(i);
        if (c == search) {
            count++;
        }
    }
    System.out.println("Search Character is-->" + search + "-> Repeated-->" + count);
}
```

6. REVERSE THE GIVEN NUMBER



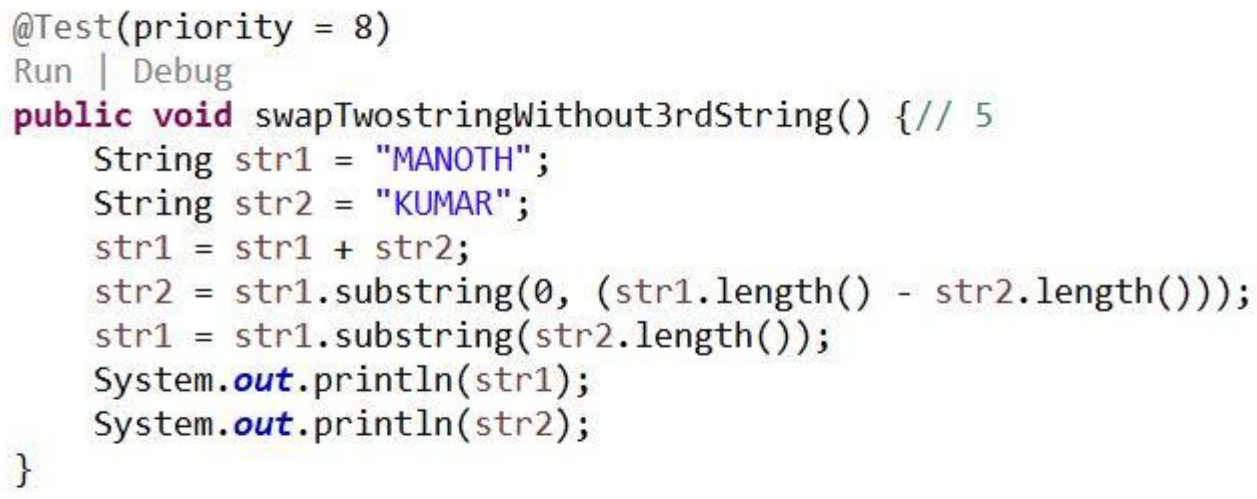
```
@Test(priority = 6)
Run | Debug
public void reverseTheGivenNumber() {
    // 4
    // System.out.println("reverseTheGivenNumber");
    long given = 7010708147l;
    long result = 0;
    while (given > 0) {
        long num = given % 10;
        result = (result * 10) + num;
        given = given / 10;
    }
    System.out.println(result);
}
```

7. CHECK GIVEN STRING IS PALINDROME



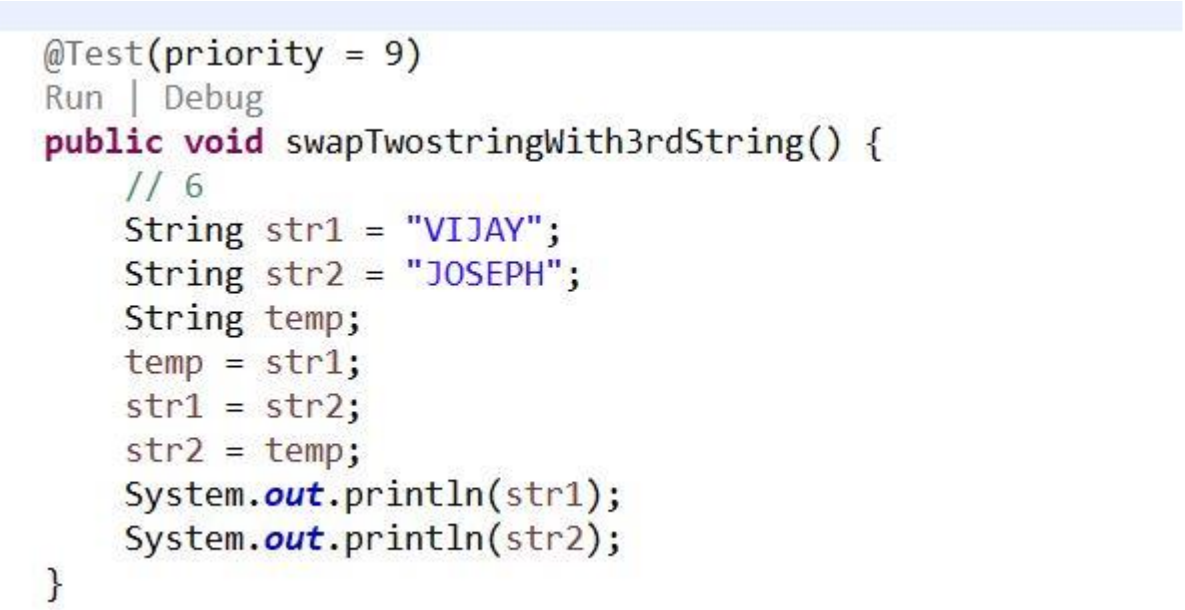
```
@Test(priority = 7)
Run | Debug
public void checkGivenStringIsPalindrom() {
    String s = "malayalam";
    String s1 = "";
    for (int i = s.length() - 1; i >= 0; i--) {
        char c = s.charAt(i);
        s1 = s1 + c;
    }
    if (s1.equals(s)) {
        System.out.println("PALIDROME");
    } else {
        System.out.println("NOT A PALINDROME");
    }
}
```

8. SWAP TWO STRINGS WITHOUT THIRD VARIABLE



```
@Test(priority = 8)
Run | Debug
public void swapTwostringWithout3rdString() {// 5
    String str1 = "MANOTH";
    String str2 = "KUMAR";
    str1 = str1 + str2;
    str2 = str1.substring(0, (str1.length() - str2.length()));
    str1 = str1.substring(str2.length());
    System.out.println(str1);
    System.out.println(str2);
}
```

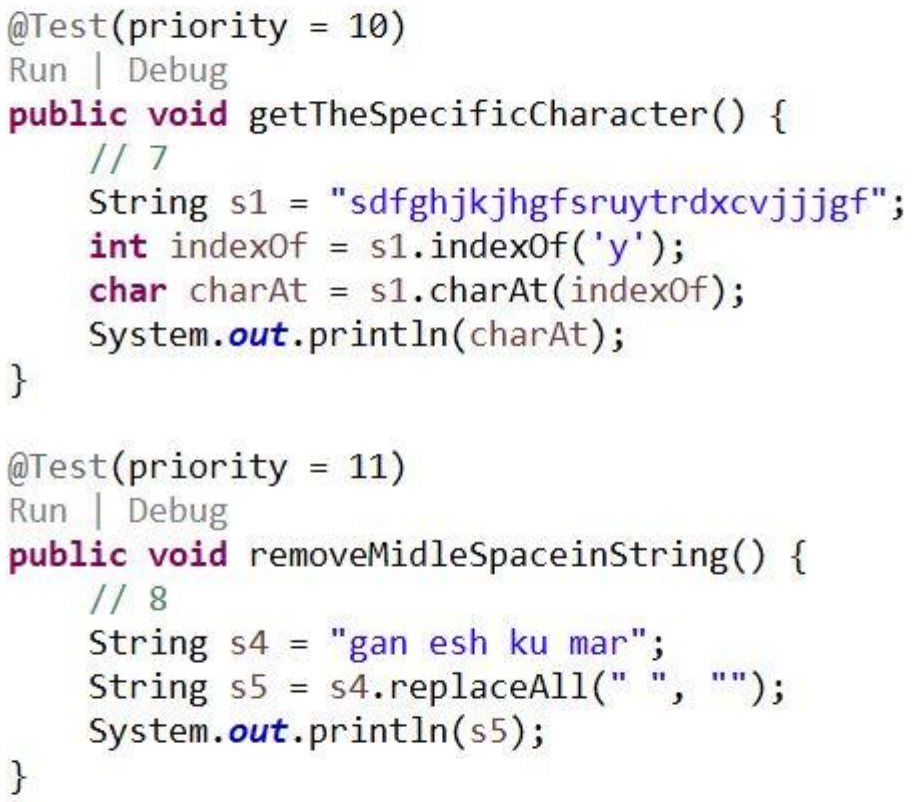
9. SWAP TWO STRINGS WITH THIRD VARIABLE



```
@Test(priority = 9)
Run | Debug
public void swapTwostringWith3rdString() {
    // 6
    String str1 = "VIJAY";
    String str2 = "JOSEPH";
    String temp;
    temp = str1;
    str1 = str2;
    str2 = temp;
    System.out.println(str1);
    System.out.println(str2);
}
```

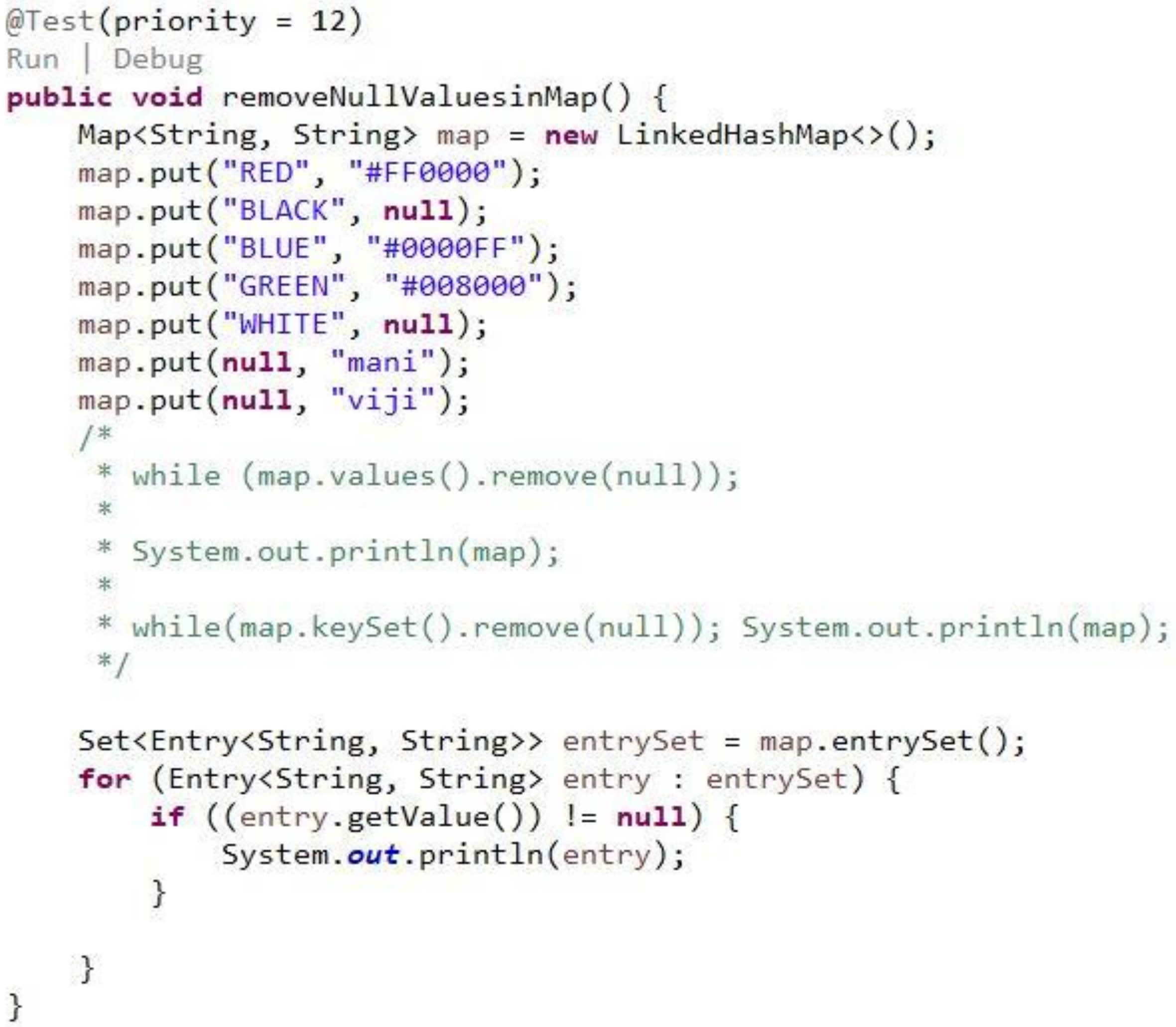
10. STRING GET SPECIFIC CHARACTER

11. REMOVE THE SPACE IN MIDDLE STRING



```
@Test(priority = 10)
Run | Debug
public void getTheSpecificCharacter() {
    // 7
    String s1 = "sdfghjkjhgfsruytrdxcvjjjgf";
    int indexOf = s1.indexOf('y');
    char charAt = s1.charAt(indexOf);
    System.out.println(charAt);
}

@Test(priority = 11)
Run | Debug
public void removeMidleSpaceinString() {
    // 8
    String s4 = "gan esh ku mar";
    String s5 = s4.replaceAll(" ", "");
    System.out.println(s5);
}
```

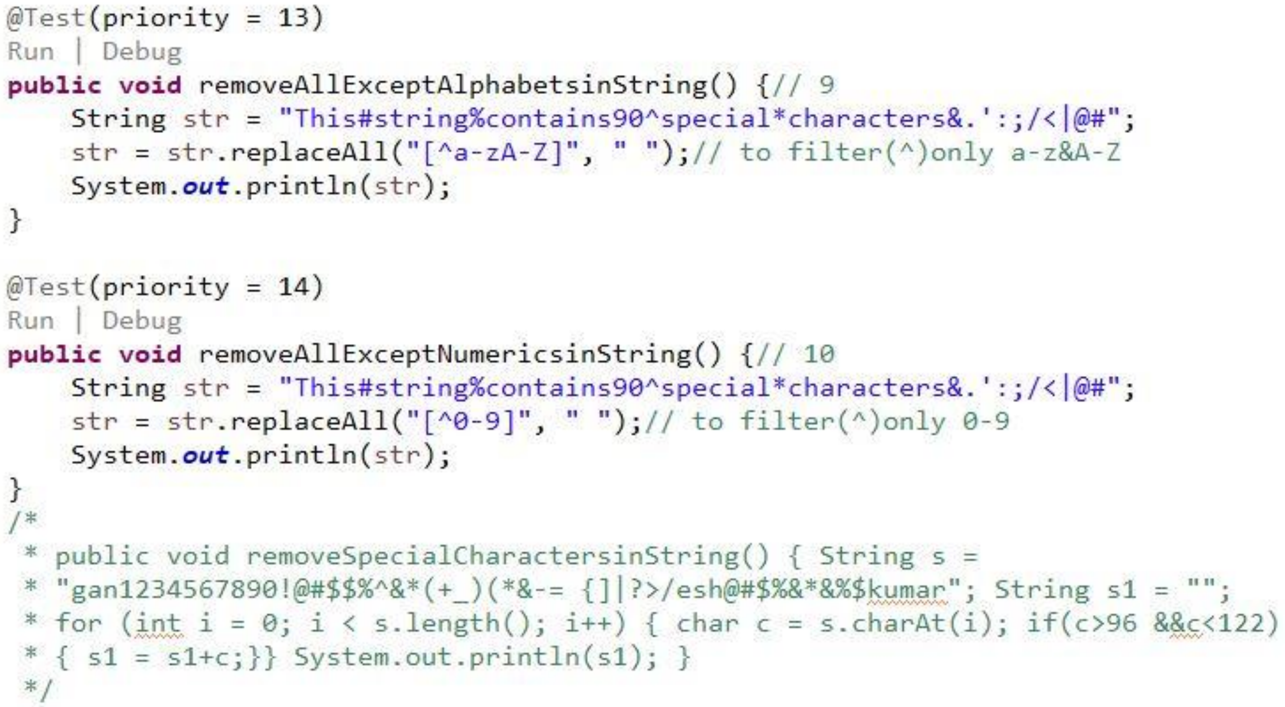
12. REMOVE NULL VALUES IN MAP



```
@Test(priority = 12)
Run | Debug
public void removeNullValuesinMap() {
    Map<String, String> map = new LinkedHashMap<>();
    map.put("RED", "#FF0000");
    map.put("BLACK", null);
    map.put("BLUE", "#0000FF");
    map.put("GREEN", "#008000");
    map.put("WHITE", null);
    map.put(null, "mani");
    map.put(null, "viji");
    /*
     * while (map.values().remove(null));
     *
     * System.out.println(map);
     *
     * while(map.keySet().remove(null)); System.out.println(map);
     */

    Set<Entry<String, String>> entrySet = map.entrySet();
    for (Entry<String, String> entry : entrySet) {
        if ((entry.getValue()) != null) {
            System.out.println(entry);
        }

    }
}
```

13. REMOVE ALL EXCEPT ALPHABETS IN STRINGS

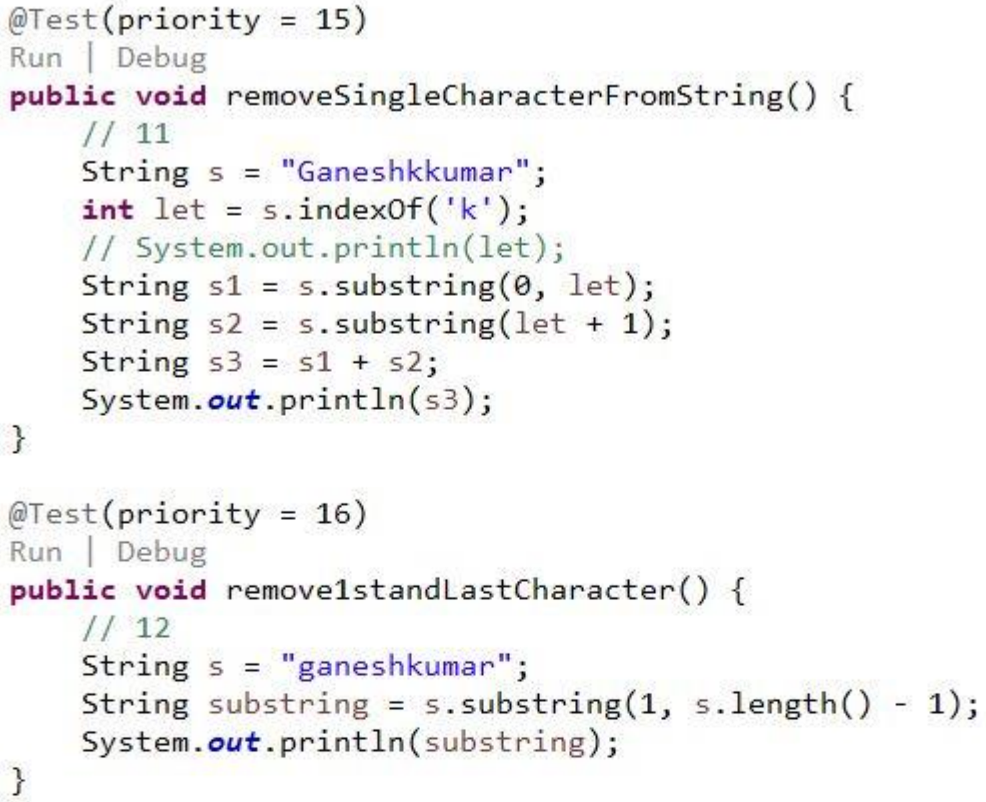
14. REMOVE ALL EXCEPT NUMERICS IN STRINGS



```
@Test(priority = 13)
Run | Debug
public void removeAllExceptAlphabetsinString() {// 9
    String str = "This#string%contains90^special*characters&.':;/<|@#";
    str = str.replaceAll("[^a-zA-Z]", " ");// to filter(^)only a-z&A-Z
    System.out.println(str);
}

@Test(priority = 14)
Run | Debug
public void removeAllExceptNumericsinString() {// 10
    String str = "This#string%contains90^special*characters&.':;/<|@#";
    str = str.replaceAll("[^0-9]", " ");// to filter(^)only 0-9
    System.out.println(str);
}
/*
 * public void removeSpecialCharactersinString() { String s =
 * "gan1234567890!@#$$%^&*(+_)(*&-= {]|?>/esh@#$%&*&%$kumar"; String s1 = "";
 * for (int i = 0; i < s.length(); i++) { char c = s.charAt(i); if(c>96 &&c<122)
 * { s1 = s1+c;}} System.out.println(s1); }
 */
```

15. REMOVE SINGLE CHARACTER IN STRING

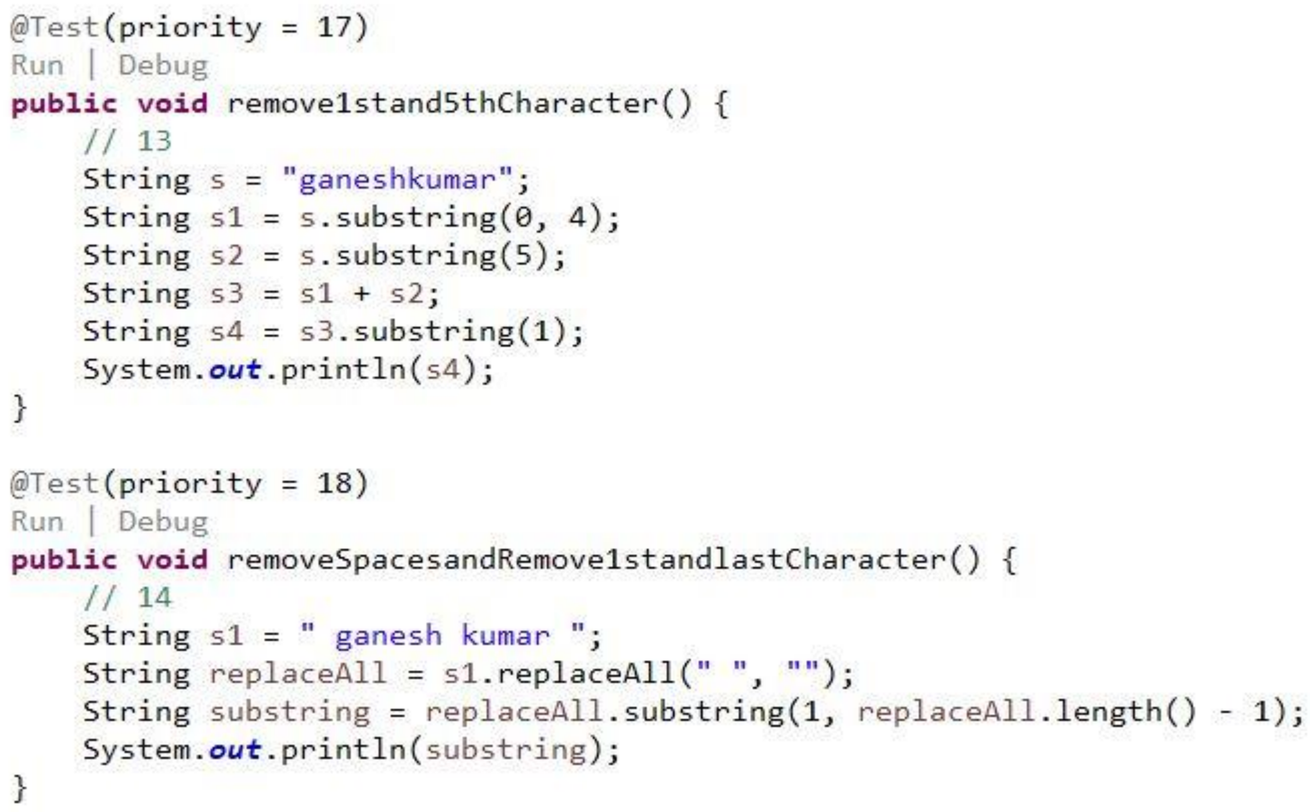
16. REMOVE 1ST AND LAST CHAR IN STRING



```
@Test(priority = 15)
Run | Debug
public void removeSingleCharacterFromString() {
    // 11
    String s = "Ganeshkkumar";
    int let = s.indexOf('k');
    // System.out.println(let);
    String s1 = s.substring(0, let);
    String s2 = s.substring(let + 1);
    String s3 = s1 + s2;
    System.out.println(s3);
}

@Test(priority = 16)
Run | Debug
public void remove1standLastCharacter() {
    // 12
    String s = "ganeshkumar";
    String substring = s.substring(1, s.length() - 1);
    System.out.println(substring);
}
```

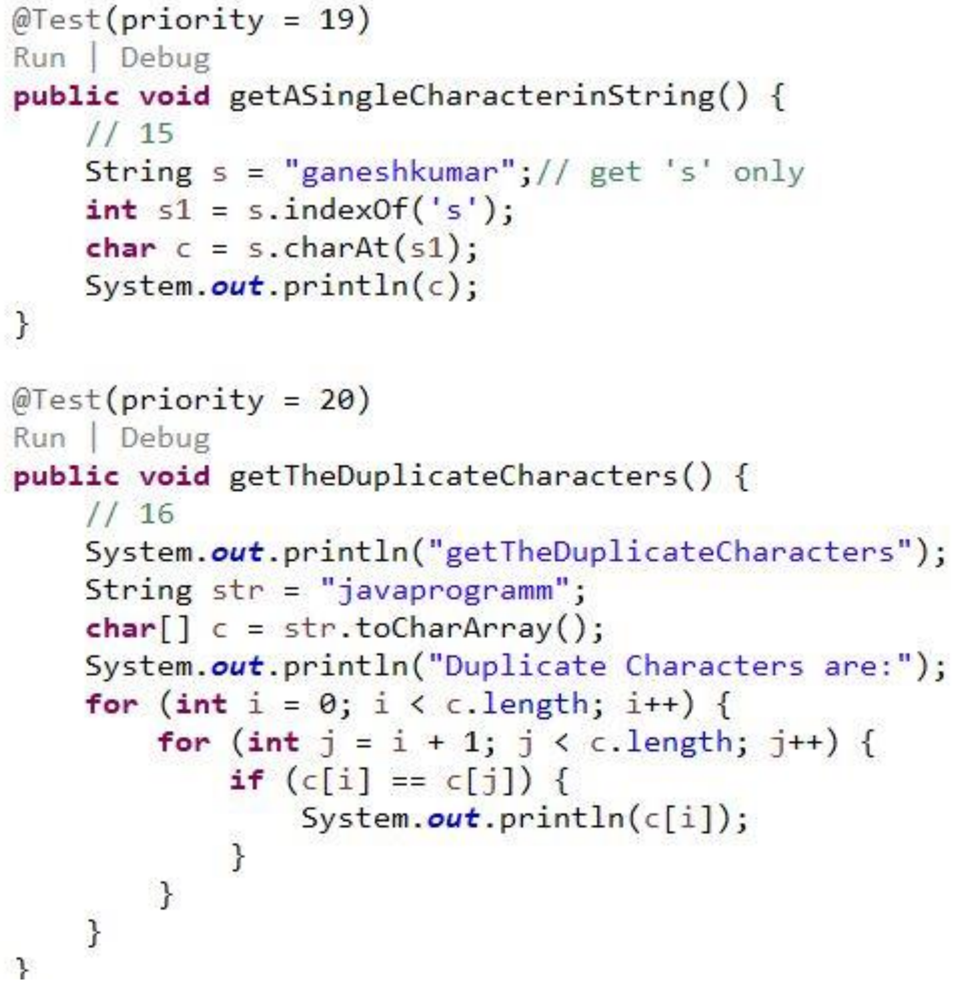
17. REMOVE 1ST AND 5TH CHAR IN STRING

18. REMOVE SPACE AND REMOVE 1ST AND LAST CHAR IN STRING



```
@Test(priority = 17)
Run | Debug
public void remove1stand5thCharacter() {
    // 13
    String s = "ganeshkumar";
    String s1 = s.substring(0, 4);
    String s2 = s.substring(5);
    String s3 = s1 + s2;
    String s4 = s3.substring(1);
    System.out.println(s4);
}

@Test(priority = 18)
Run | Debug
public void removeSpacesandRemove1standlastCharacter() {
    // 14
    String s1 = " ganesh kumar ";
    String replaceAll = s1.replaceAll(" ", "");
    String substring = replaceAll.substring(1, replaceAll.length() - 1);
    System.out.println(substring);
}
```

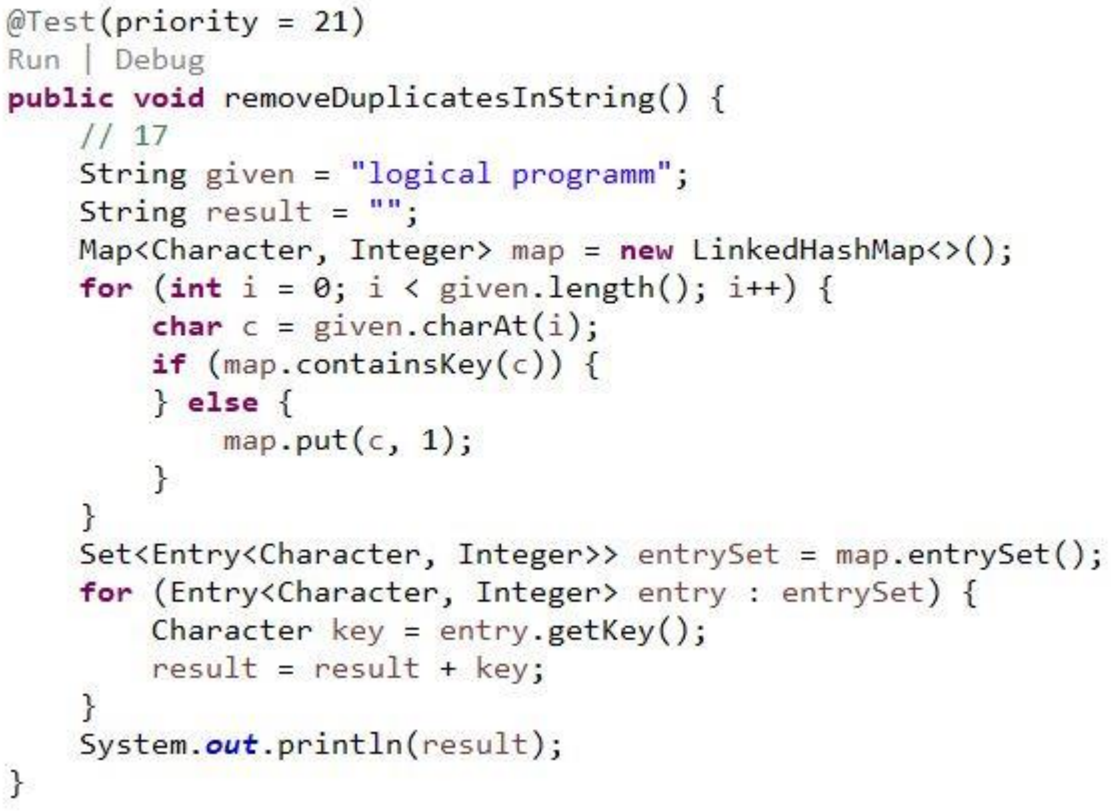
19. GET SINGLE CHAR IN STRING
20. GET DUPLICATES CHAR IN STRING



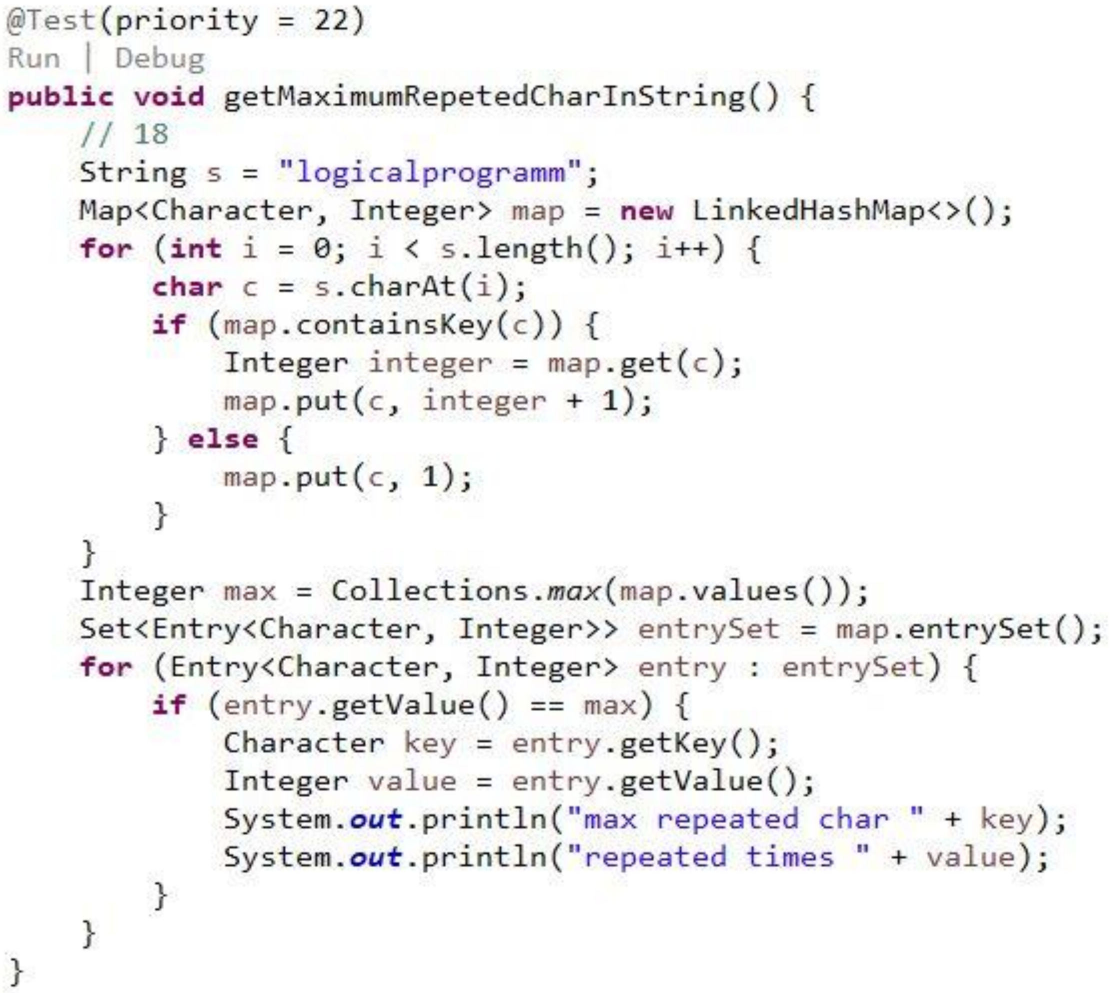
```
@Test(priority = 19)
Run | Debug
public void getASingleCharacterinString() {
    // 15
    String s = "ganeshkumar";// get 's' only
    int s1 = s.indexOf('s');
    char c = s.charAt(s1);
    System.out.println(c);
}

@Test(priority = 20)
Run | Debug
public void getTheDuplicateCharacters() {
    // 16
    System.out.println("getTheDuplicateCharacters");
    String str = "javaprogramm";
    char[] c = str.toCharArray();
    System.out.println("Duplicate Characters are:");
    for (int i = 0; i < c.length; i++) {
        for (int j = i + 1; j < c.length; j++) {
            if (c[i] == c[j]) {
                System.out.println(c[i]);
            }
        }
    }
}
```

21. REMOVE DUPLICATE IN STRING

22. GET THE MAXIMUM REPEATED CHAR IN STRING

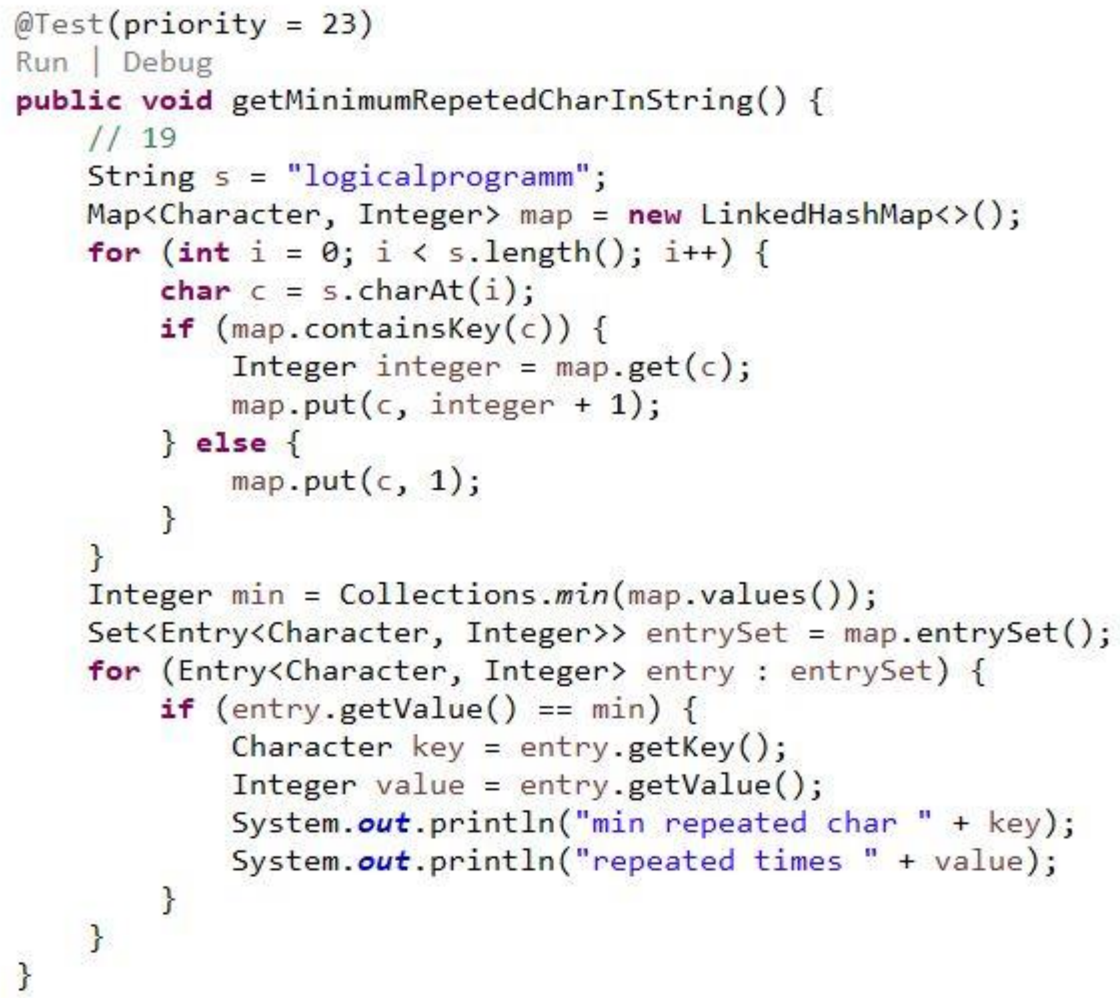


```java
@Test(priority = 21)
Run | Debug
public void removeDuplicatesInString() {
    // 17
    String given = "logical programm";
    String result = "";
    Map<Character, Integer> map = new LinkedHashMap<>();
    for (int i = 0; i < given.length(); i++) {
        char c = given.charAt(i);
        if (map.containsKey(c)) {
        } else {
            map.put(c, 1);
        }
    }
    Set<Entry<Character, Integer>> entrySet = map.entrySet();
    for (Entry<Character, Integer> entry : entrySet) {
        Character key = entry.getKey();
        result = result + key;
    }
    System.out.println(result);
}
```



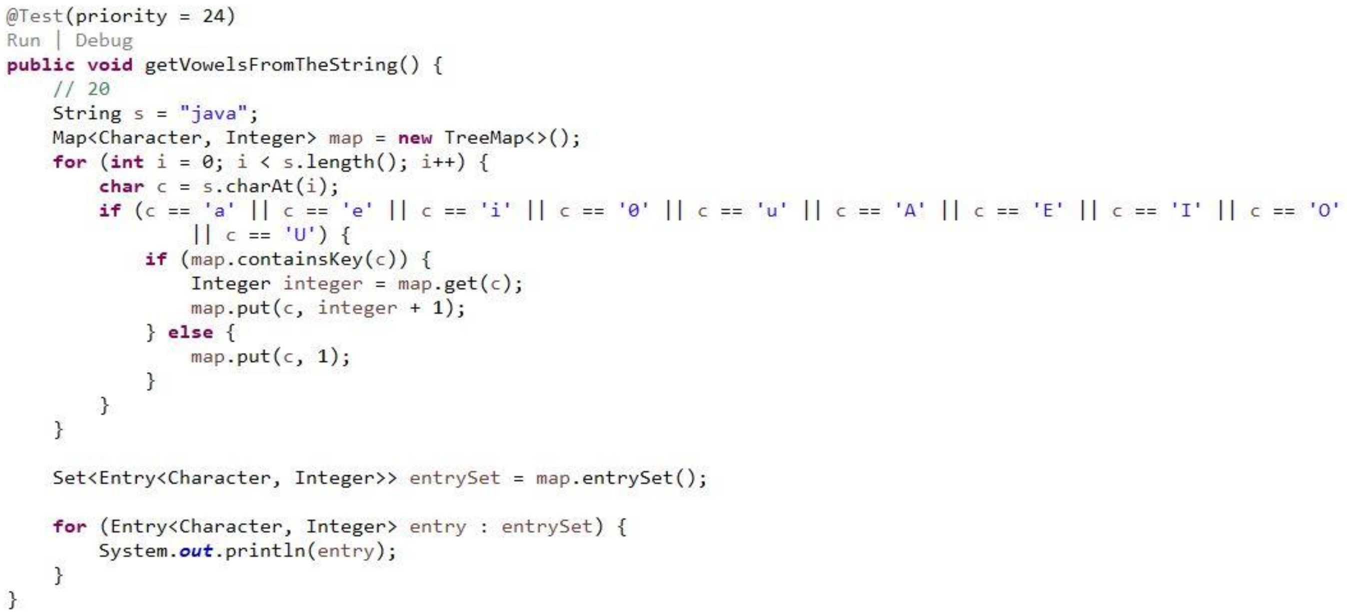
```
@Test(priority = 22)
Run | Debug
public void getMaximumRepetedCharInString() {
    // 18
    String s = "logicalprogramm";
    Map<Character, Integer> map = new LinkedHashMap<>();
    for (int i = 0; i < s.length(); i++) {
        char c = s.charAt(i);
        if (map.containsKey(c)) {
            Integer integer = map.get(c);
            map.put(c, integer + 1);
        } else {
            map.put(c, 1);
        }
    }
    Integer max = Collections.max(map.values());
    Set<Entry<Character, Integer>> entrySet = map.entrySet();
    for (Entry<Character, Integer> entry : entrySet) {
        if (entry.getValue() == max) {
            Character key = entry.getKey();
            Integer value = entry.getValue();
            System.out.println("max repeated char " + key);
            System.out.println("repeated times " + value);
        }
    }
}
```

23. GET THE MINIMUM REPEATED CHAR IN STRING



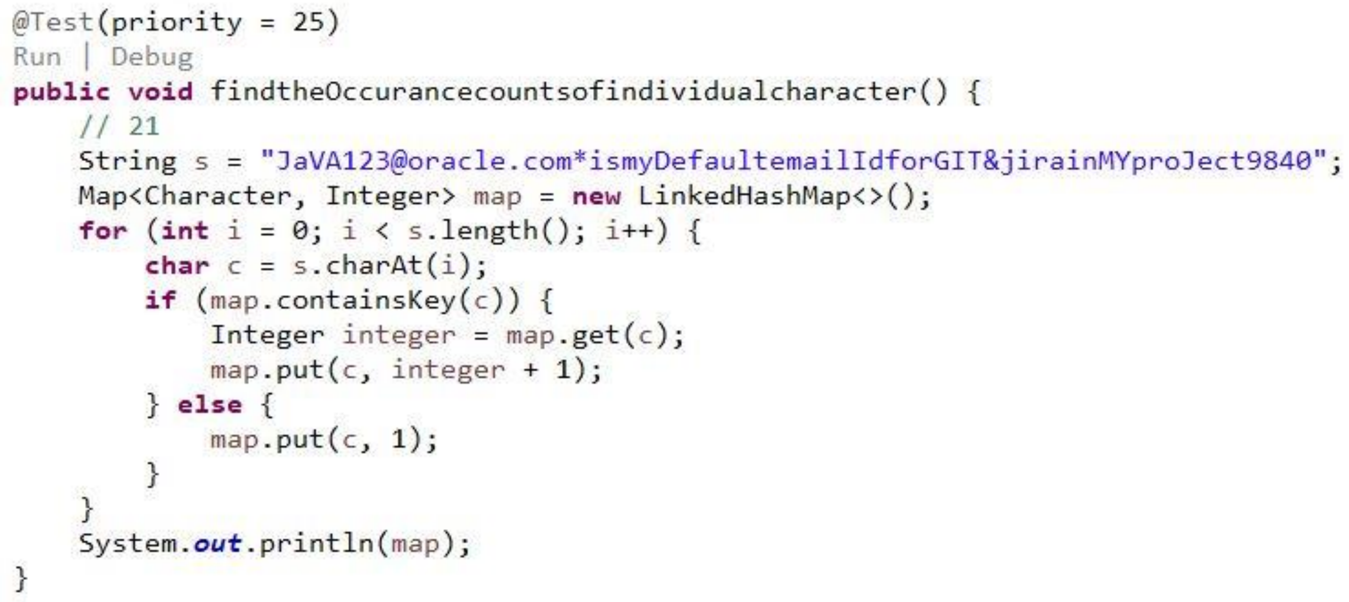
```
@Test(priority = 23)
Run | Debug
public void getMinimumRepetedCharInString() {
    // 19
    String s = "logicalprogramm";
    Map<Character, Integer> map = new LinkedHashMap<>();
    for (int i = 0; i < s.length(); i++) {
        char c = s.charAt(i);
        if (map.containsKey(c)) {
            Integer integer = map.get(c);
            map.put(c, integer + 1);
        } else {
            map.put(c, 1);
        }
    }
    Integer min = Collections.min(map.values());
    Set<Entry<Character, Integer>> entrySet = map.entrySet();
    for (Entry<Character, Integer> entry : entrySet) {
        if (entry.getValue() == min) {
            Character key = entry.getKey();
            Integer value = entry.getValue();
            System.out.println("min repeated char " + key);
            System.out.println("repeated times " + value);
        }
    }
}
```

24. GET VOWELS IN STRING



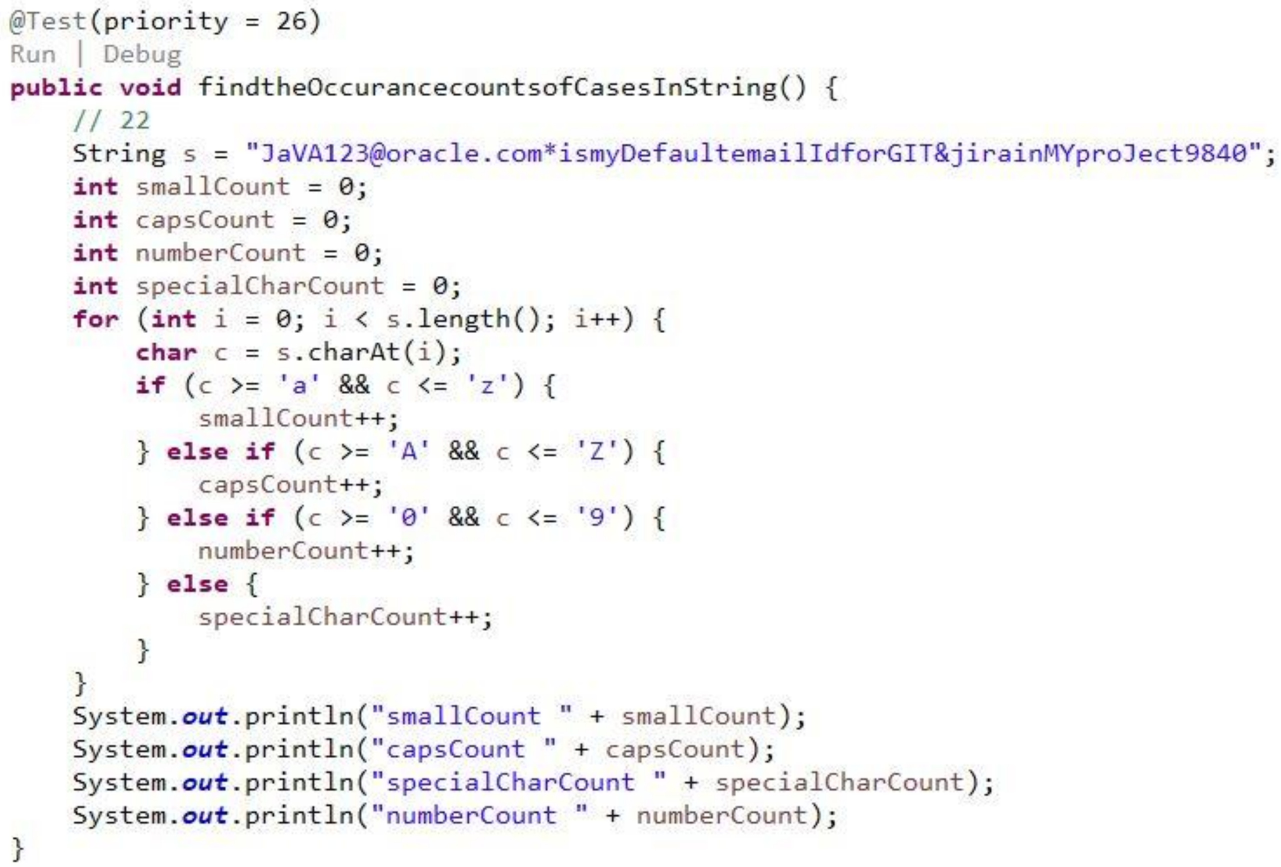
```
@Test(priority = 24)
Run | Debug
public void getVowelsFromTheString() {
    // 20
    String s = "java";
    Map<Character, Integer> map = new TreeMap<>();
    for (int i = 0; i < s.length(); i++) {
        char c = s.charAt(i);
        if (c == 'a' || c == 'e' || c == 'i' || c == '0' || c == 'u' || c == 'A' || c == 'E' || c == 'I' || c == 'O'
                || c == 'U') {
            if (map.containsKey(c)) {
                Integer integer = map.get(c);
                map.put(c, integer + 1);
            } else {
                map.put(c, 1);
            }
        }
    }

    Set<Entry<Character, Integer>> entrySet = map.entrySet();

    for (Entry<Character, Integer> entry : entrySet) {
        System.out.println(entry);
    }
}
```

25. FIND THE OCCURRENCE AND COUNT OF EACH CHAR



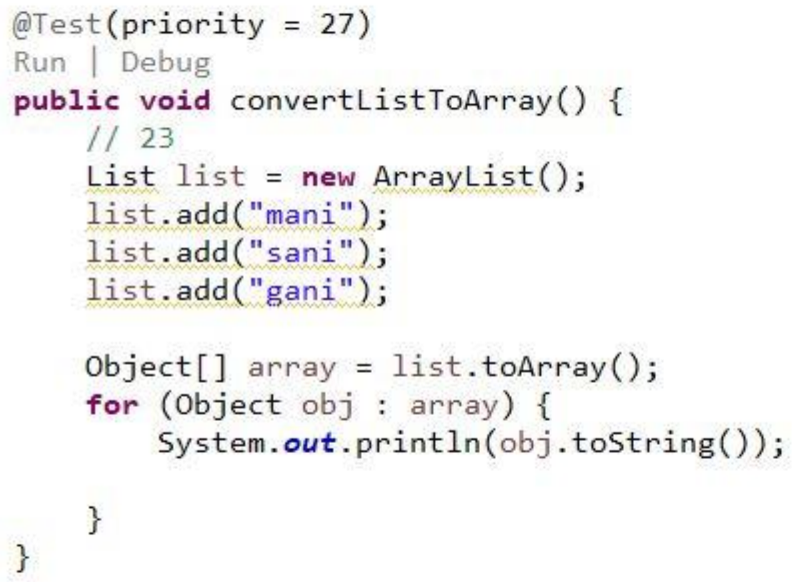
```
@Test(priority = 25)
Run | Debug
public void findtheOccurancecountsofindividualcharacter() {
    // 21
    String s = "JaVA123@oracle.com*ismyDefaultemailIdforGIT&jirainMYproJect9840";
    Map<Character, Integer> map = new LinkedHashMap<>();
    for (int i = 0; i < s.length(); i++) {
        char c = s.charAt(i);
        if (map.containsKey(c)) {
            Integer integer = map.get(c);
            map.put(c, integer + 1);
        } else {
            map.put(c, 1);
        }
    }
    System.out.println(map);
}
```

26. FIND THE COUNT OF DIFFERENT CASE IN STRING



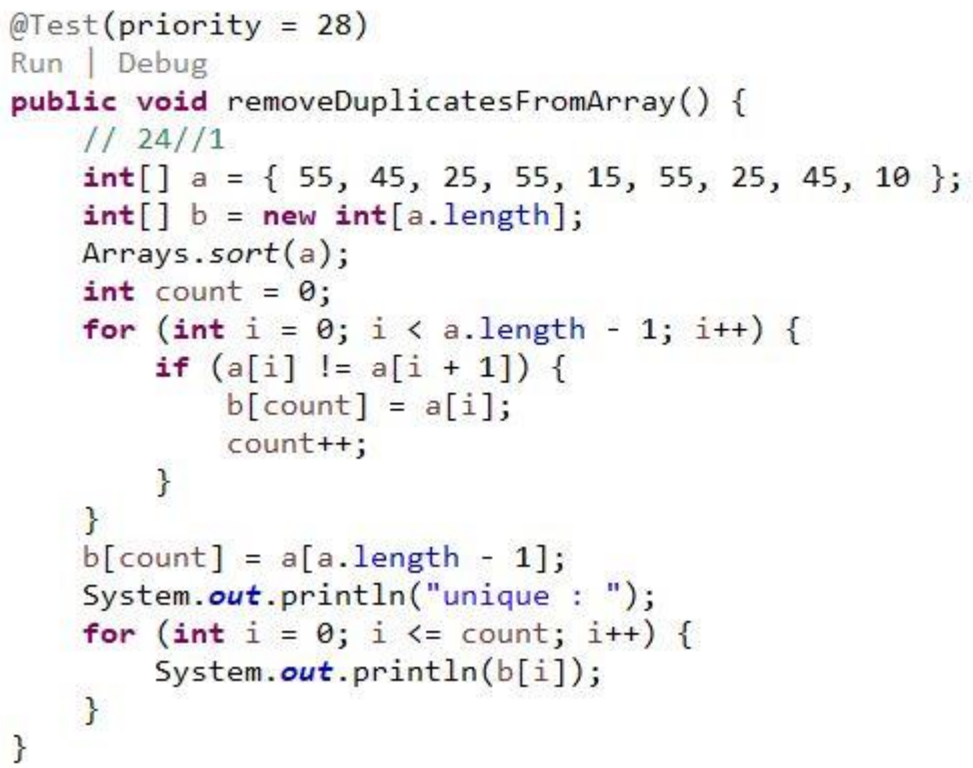
```
@Test(priority = 26)
Run | Debug
public void findtheOccurancecountsofCasesInString() {
    // 22
    String s = "JaVA123@oracle.com*ismyDefaultemailIdforGIT&jirainMYproJect9840";
    int smallCount = 0;
    int capsCount = 0;
    int numberCount = 0;
    int specialCharCount = 0;
    for (int i = 0; i < s.length(); i++) {
        char c = s.charAt(i);
        if (c >= 'a' && c <= 'z') {
            smallCount++;
        } else if (c >= 'A' && c <= 'Z') {
            capsCount++;
        } else if (c >= '0' && c <= '9') {
            numberCount++;
        } else {
            specialCharCount++;
        }
    }
    System.out.println("smallCount " + smallCount);
    System.out.println("capsCount " + capsCount);
    System.out.println("specialCharCount " + specialCharCount);
    System.out.println("numberCount " + numberCount);
}
```

27. CONVERT LIST TO ARRAY



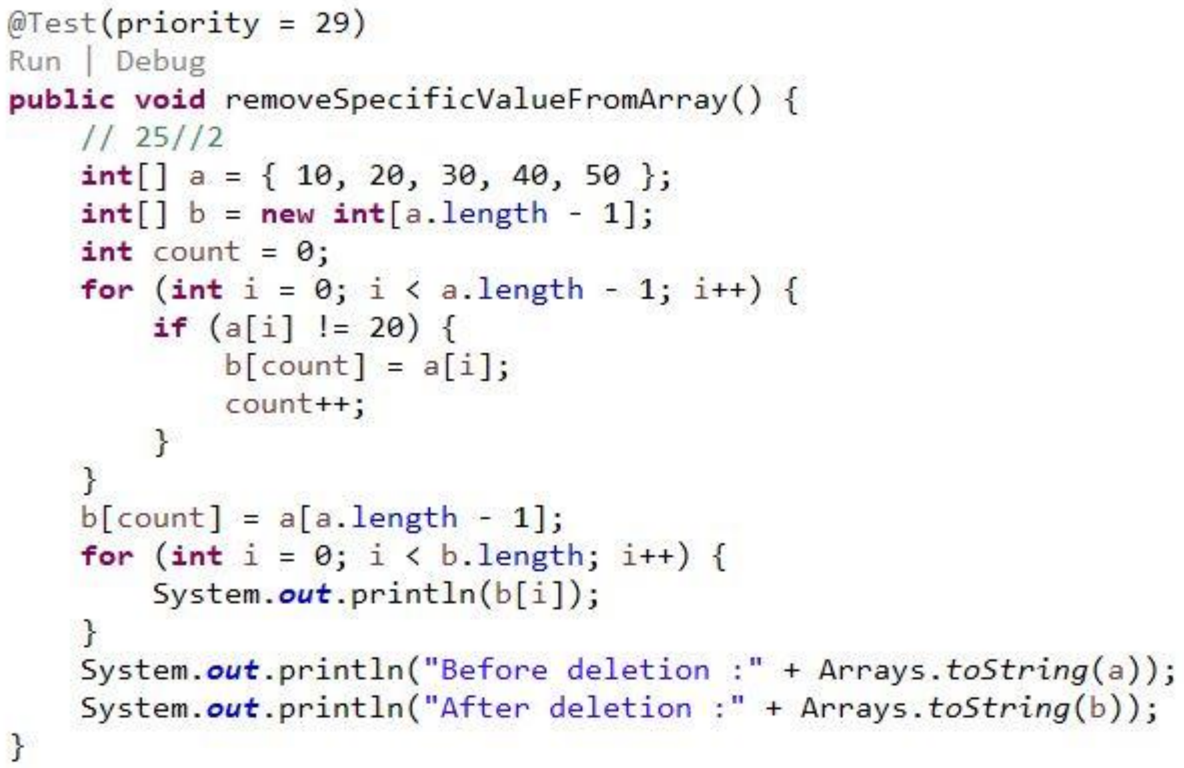
```
@Test(priority = 27)
Run | Debug
public void convertListToArray() {
    // 23
    List list = new ArrayList();
    list.add("mani");
    list.add("sani");
    list.add("gani");

    Object[] array = list.toArray();
    for (Object obj : array) {
        System.out.println(obj.toString());

    }
}
```

28. REMOVE DUPLICATES IN ARRAY



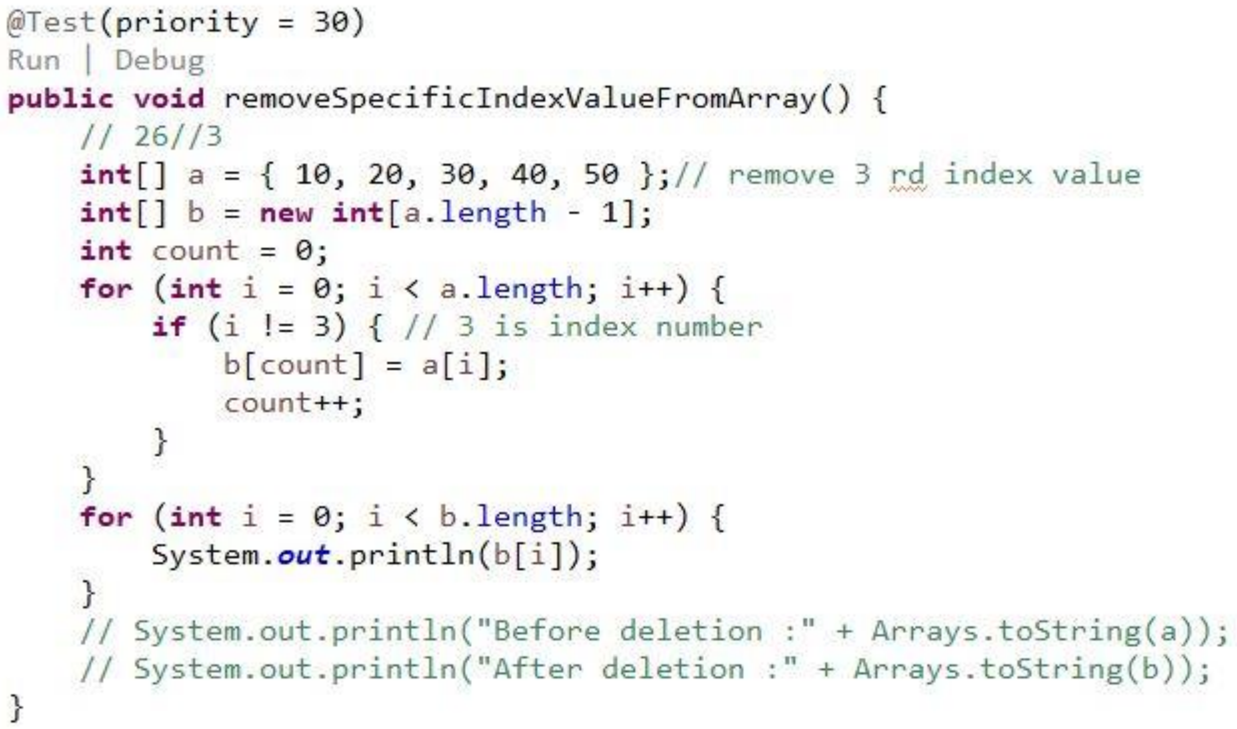
```
@Test(priority = 28)
Run | Debug
public void removeDuplicatesFromArray() {
    // 24//1
    int[] a = { 55, 45, 25, 55, 15, 55, 25, 45, 10 };
    int[] b = new int[a.length];
    Arrays.sort(a);
    int count = 0;
    for (int i = 0; i < a.length - 1; i++) {
        if (a[i] != a[i + 1]) {
            b[count] = a[i];
            count++;
        }
    }
    b[count] = a[a.length - 1];
    System.out.println("unique : ");
    for (int i = 0; i <= count; i++) {
        System.out.println(b[i]);
    }
}
```

29. REMOVE SPECIFIC VALUE FROM ARRAY



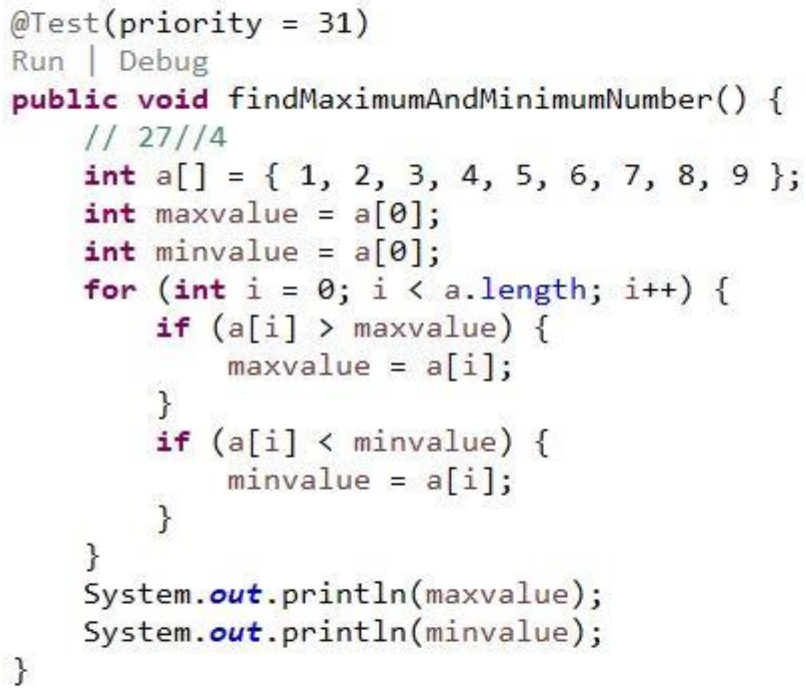
```java
@Test(priority = 29)
Run | Debug
public void removeSpecificValueFromArray() {
    // 25//2
    int[] a = { 10, 20, 30, 40, 50 };
    int[] b = new int[a.length - 1];
    int count = 0;
    for (int i = 0; i < a.length - 1; i++) {
        if (a[i] != 20) {
            b[count] = a[i];
            count++;
        }
    }
    b[count] = a[a.length - 1];
    for (int i = 0; i < b.length; i++) {
        System.out.println(b[i]);
    }
    System.out.println("Before deletion :" + Arrays.toString(a));
    System.out.println("After deletion :" + Arrays.toString(b));
}
```

30. REMOVE SPECIFIC INDEX VALUE IN ARRAY



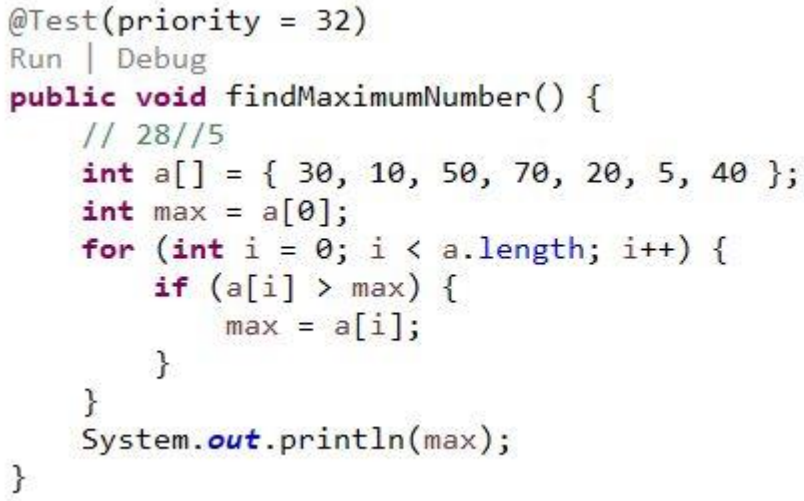
```java
@Test(priority = 30)
Run | Debug
public void removeSpecificIndexValueFromArray() {
    // 26//3
    int[] a = { 10, 20, 30, 40, 50 };// remove 3 rd index value
    int[] b = new int[a.length - 1];
    int count = 0;
    for (int i = 0; i < a.length; i++) {
        if (i != 3) { // 3 is index number
            b[count] = a[i];
            count++;
        }
    }
    for (int i = 0; i < b.length; i++) {
        System.out.println(b[i]);
    }
    // System.out.println("Before deletion :" + Arrays.toString(a));
    // System.out.println("After deletion :" + Arrays.toString(b));
}
```

31. FIND THE MAX AND MIN NUMBER IN ARRAY



```
@Test(priority = 31)
Run | Debug
public void findMaximumAndMinimumNumber() {
    // 27//4
    int a[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9 };
    int maxvalue = a[0];
    int minvalue = a[0];
    for (int i = 0; i < a.length; i++) {
        if (a[i] > maxvalue) {
            maxvalue = a[i];
        }
        if (a[i] < minvalue) {
            minvalue = a[i];
        }
    }
    System.out.println(maxvalue);
    System.out.println(minvalue);
}
```

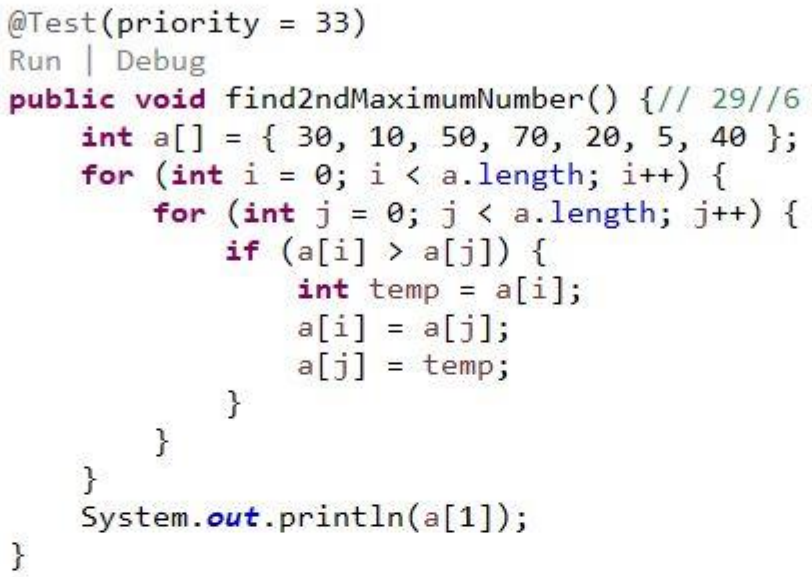
32. FIND MAXIMUM NUMBER IN ARRAY



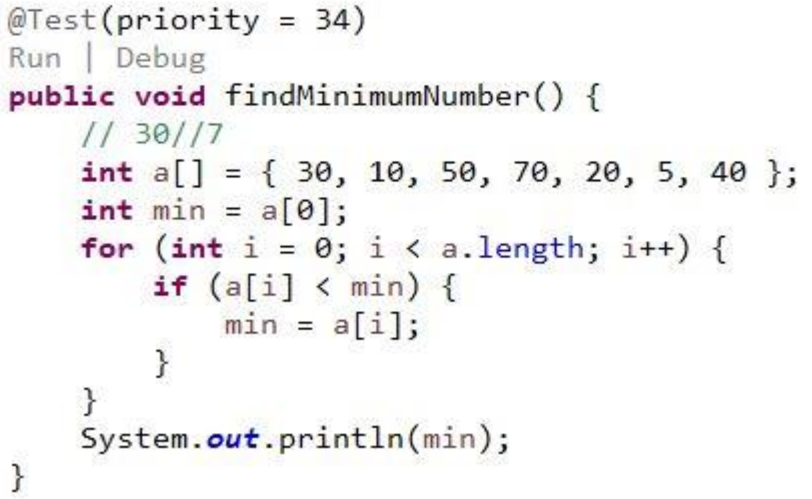
```
@Test(priority = 32)
Run | Debug
public void findMaximumNumber() {
    // 28//5
    int a[] = { 30, 10, 50, 70, 20, 5, 40 };
    int max = a[0];
    for (int i = 0; i < a.length; i++) {
        if (a[i] > max) {
            max = a[i];
        }
    }
    System.out.println(max);
}
```

33. FIND 2ND MAXIMUM NUM IN ARRAY

34. FIND THE MINIMUM NUM IN ARRAY

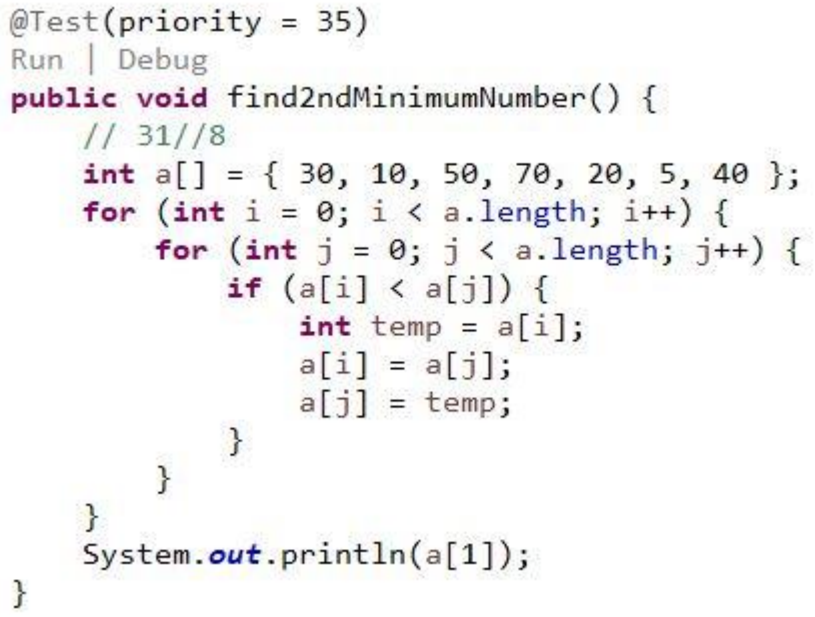


```
@Test(priority = 33)
Run | Debug
public void find2ndMaximumNumber() {// 29//6
    int a[] = { 30, 10, 50, 70, 20, 5, 40 };
    for (int i = 0; i < a.length; i++) {
        for (int j = 0; j < a.length; j++) {
            if (a[i] > a[j]) {
                int temp = a[i];
                a[i] = a[j];
                a[j] = temp;
            }
        }
    }
    System.out.println(a[1]);
}
```



```
@Test(priority = 34)
Run | Debug
public void findMinimumNumber() {
    // 30//7
    int a[] = { 30, 10, 50, 70, 20, 5, 40 };
    int min = a[0];
    for (int i = 0; i < a.length; i++) {
        if (a[i] < min) {
            min = a[i];
        }
    }
    System.out.println(min);
}
```

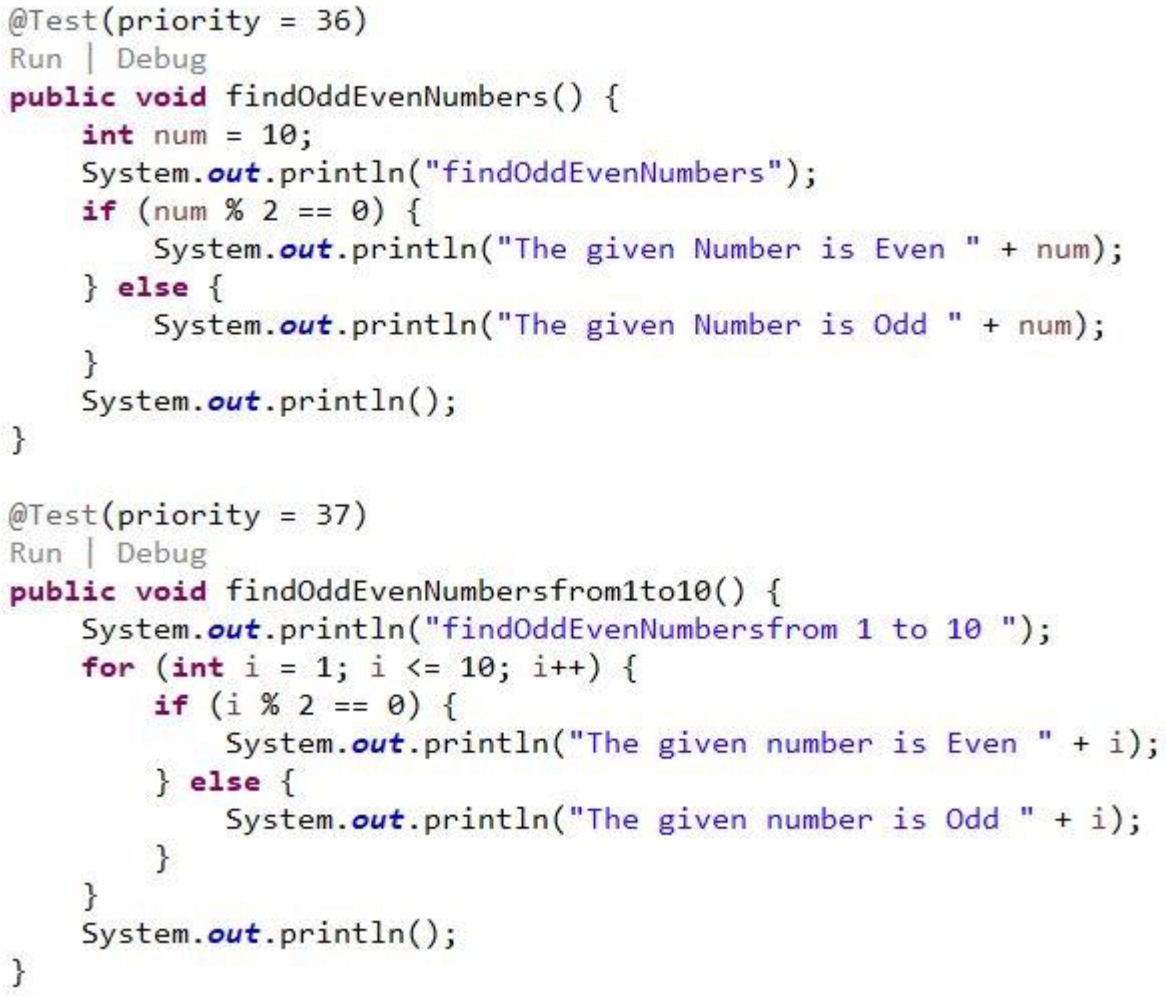
35. FIND THE 2ND MINIMUM NUM IN ARRAY



```
@Test(priority = 35)
Run | Debug
public void find2ndMinimumNumber() {
    // 31//8
    int a[] = { 30, 10, 50, 70, 20, 5, 40 };
    for (int i = 0; i < a.length; i++) {
        for (int j = 0; j < a.length; j++) {
            if (a[i] < a[j]) {
                int temp = a[i];
                a[i] = a[j];
                a[j] = temp;
            }
        }
    }
    System.out.println(a[1]);
}
```

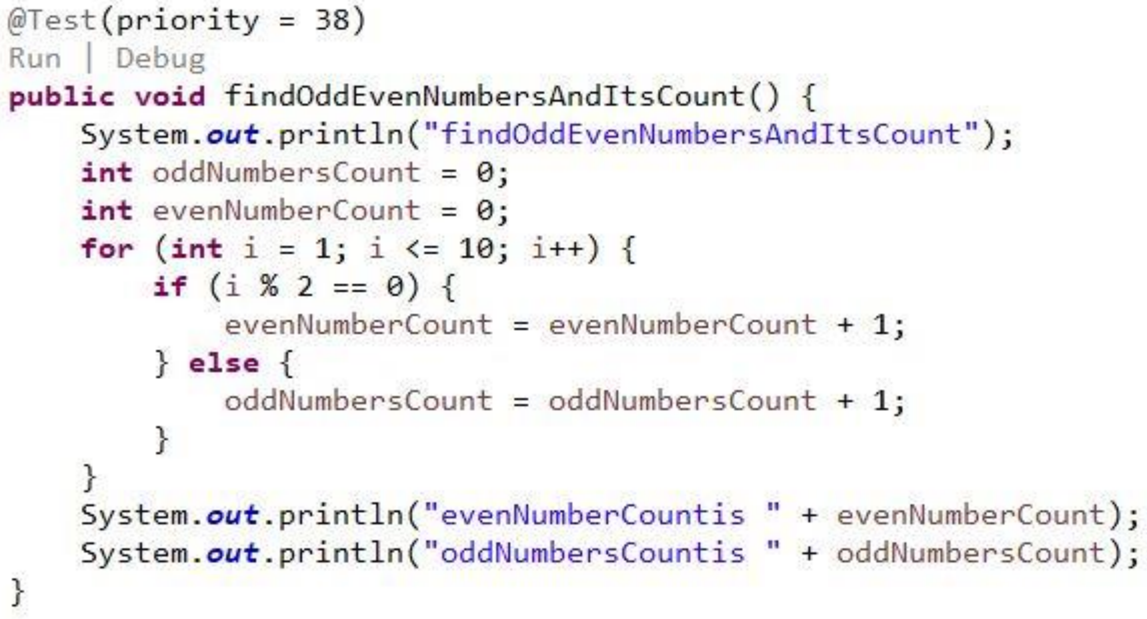
36. FIND THE ODD EVEN NUMBER

37. FIND THE ODD NUMBERS FROM 1 TO 100



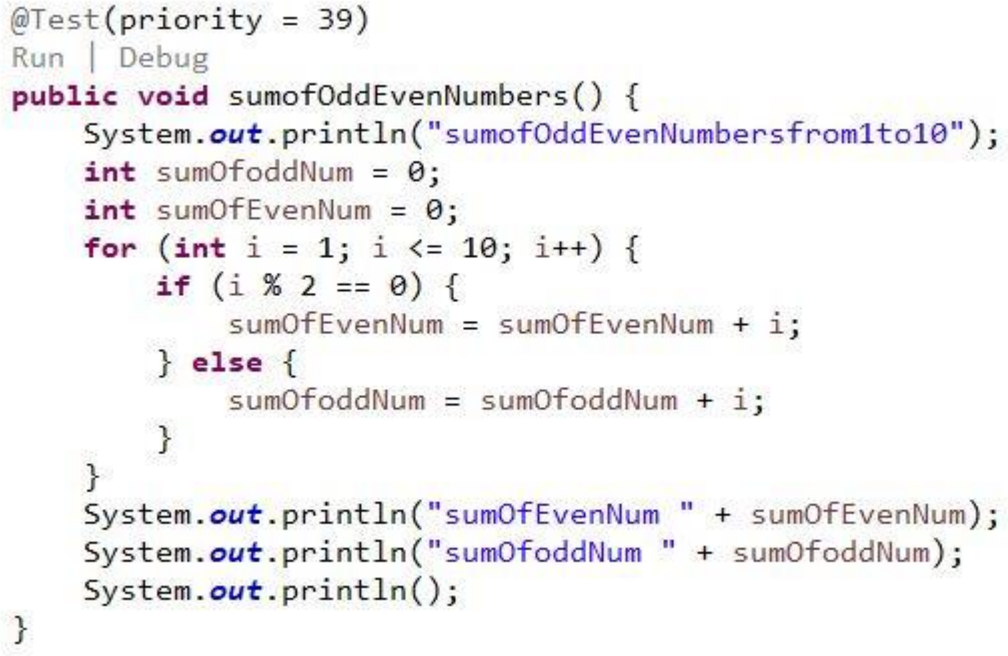
```
@Test(priority = 36)
Run | Debug
public void findOddEvenNumbers() {
    int num = 10;
    System.out.println("findOddEvenNumbers");
    if (num % 2 == 0) {
        System.out.println("The given Number is Even " + num);
    } else {
        System.out.println("The given Number is Odd " + num);
    }
    System.out.println();
}

@Test(priority = 37)
Run | Debug
public void findOddEvenNumbersfrom1to10() {
    System.out.println("findOddEvenNumbersfrom 1 to 10 ");
    for (int i = 1; i <= 10; i++) {
        if (i % 2 == 0) {
            System.out.println("The given number is Even " + i);
        } else {
            System.out.println("The given number is Odd " + i);
        }
    }
    System.out.println();
}
```

38. FIND THE ODD EVEN NUMBERS AND ITS COUNT



```
@Test(priority = 38)
Run | Debug
public void findOddEvenNumbersAndItsCount() {
    System.out.println("findOddEvenNumbersAndItsCount");
    int oddNumbersCount = 0;
    int evenNumberCount = 0;
    for (int i = 1; i <= 10; i++) {
        if (i % 2 == 0) {
            evenNumberCount = evenNumberCount + 1;
        } else {
            oddNumbersCount = oddNumbersCount + 1;
        }
    }
    System.out.println("evenNumberCountis " + evenNumberCount);
    System.out.println("oddNumbersCountis " + oddNumbersCount);
}
```

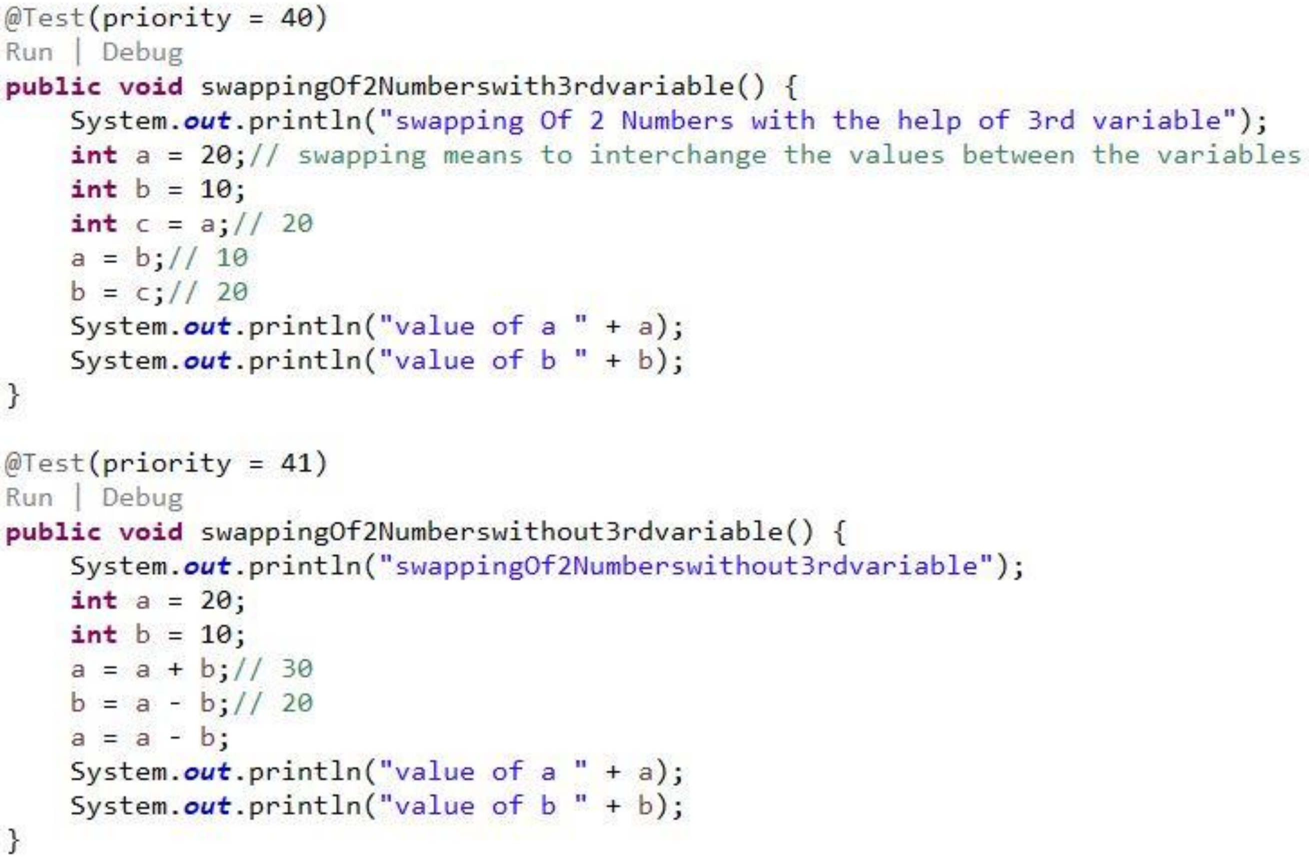
39. SUM OF ODD EVEN NUMBERS



```
@Test(priority = 39)
Run | Debug
public void sumofOddEvenNumbers() {
    System.out.println("sumofOddEvenNumbersfrom1to10");
    int sumOfoddNum = 0;
    int sumOfEvenNum = 0;
    for (int i = 1; i <= 10; i++) {
        if (i % 2 == 0) {
            sumOfEvenNum = sumOfEvenNum + i;
        } else {
            sumOfoddNum = sumOfoddNum + i;
        }
    }
    System.out.println("sumOfEvenNum " + sumOfEvenNum);
    System.out.println("sumOfoddNum " + sumOfoddNum);
    System.out.println();
}
```

40. SWAPPING THE TWO NUMBERS WITH 3RD VARIABLE

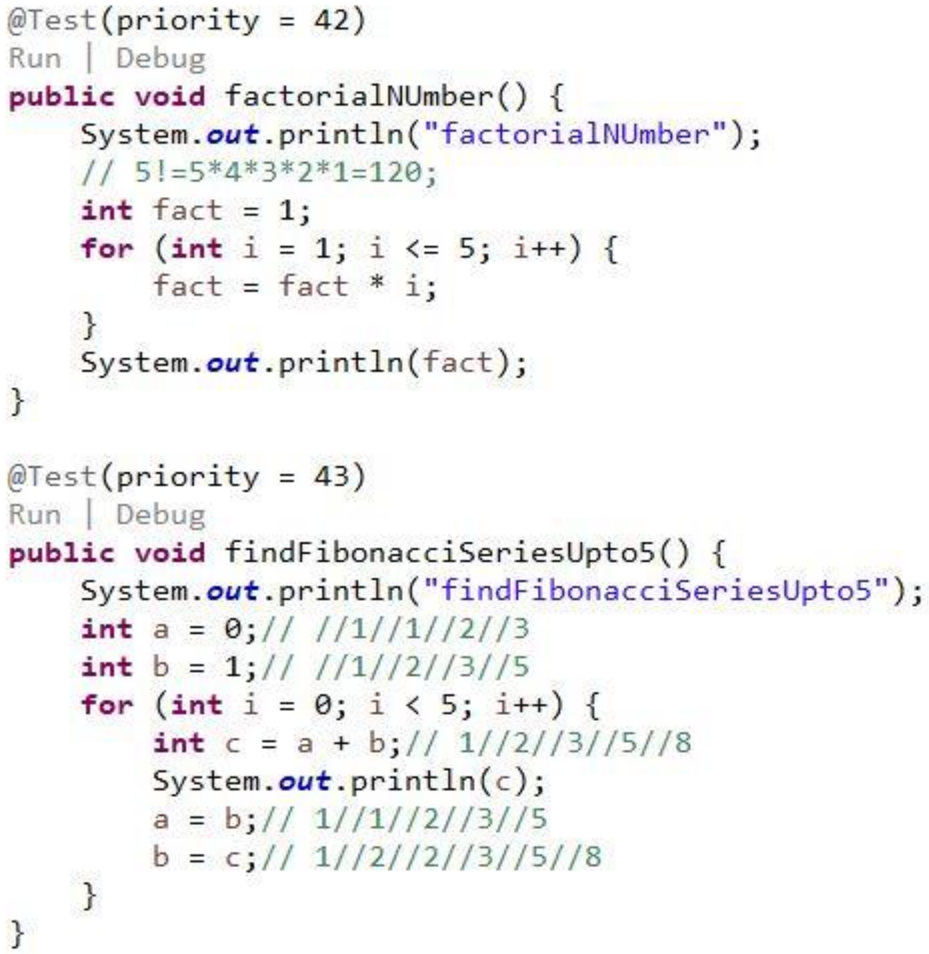
41. SWAPPING THE TWO NUMBERS WITHOUT 3RD VARIABLE



```
@Test(priority = 40)
Run | Debug
public void swappingOf2Numberswith3rdvariable() {
    System.out.println("swapping Of 2 Numbers with the help of 3rd variable");
    int a = 20;// swapping means to interchange the values between the variables
    int b = 10;
    int c = a;// 20
    a = b;// 10
    b = c;// 20
    System.out.println("value of a " + a);
    System.out.println("value of b " + b);
}

@Test(priority = 41)
Run | Debug
public void swappingOf2Numberswithout3rdvariable() {
    System.out.println("swappingOf2Numberswithout3rdvariable");
    int a = 20;
    int b = 10;
    a = a + b;// 30
    b = a - b;// 20
    a = a - b;
    System.out.println("value of a " + a);
    System.out.println("value of b " + b);
}
```

42. FACTORIAL NUMBERS

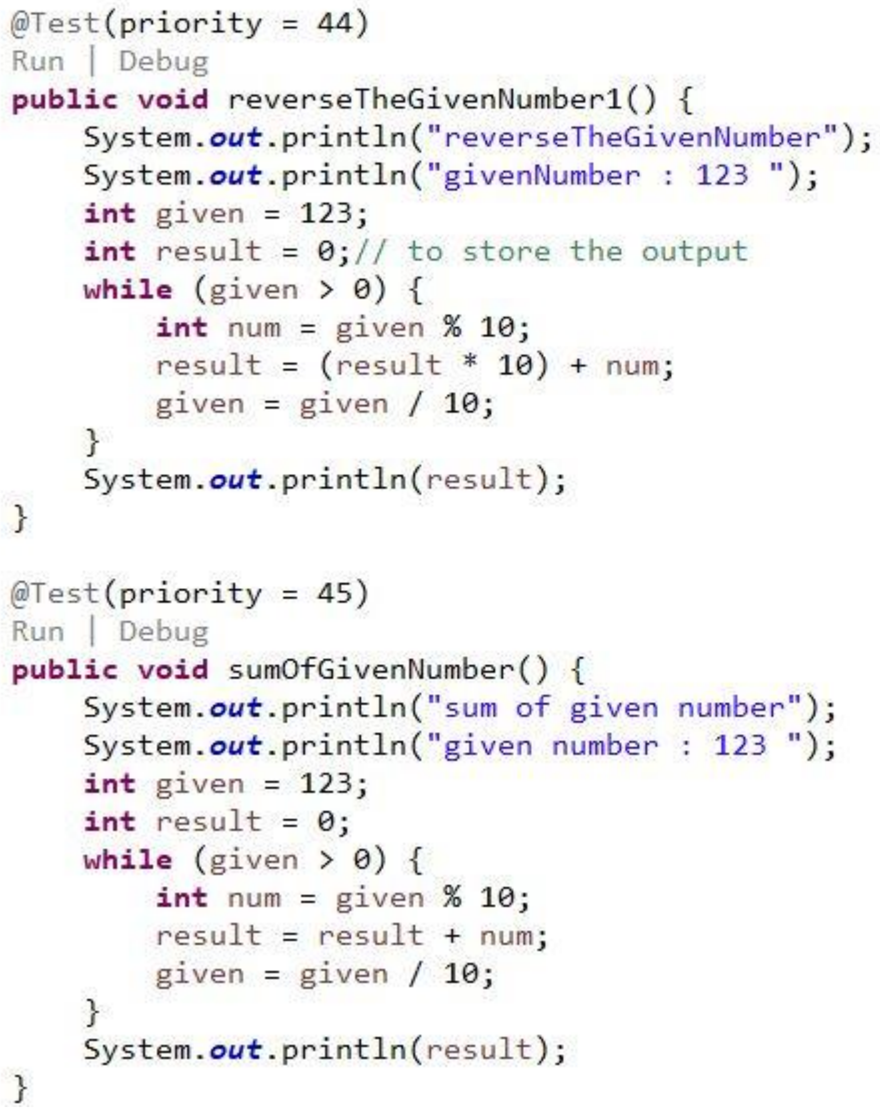
43. FIBONACCI SERIES



```
@Test(priority = 42)
Run | Debug
public void factorialNUmber() {
    System.out.println("factorialNUmber");
    // 5!=5*4*3*2*1=120;
    int fact = 1;
    for (int i = 1; i <= 5; i++) {
        fact = fact * i;
    }
    System.out.println(fact);
}

@Test(priority = 43)
Run | Debug
public void findFibonacciSeriesUpto5() {
    System.out.println("findFibonacciSeriesUpto5");
    int a = 0;// //1//1//2//3
    int b = 1;// //1//2//3//5
    for (int i = 0; i < 5; i++) {
        int c = a + b;// 1//2//3//5//8
        System.out.println(c);
        a = b;// 1//1//2//3//5
        b = c;// 1//2//2//3//5//8
    }
}
```

44. REVERSE THE GIVEN NUMBER

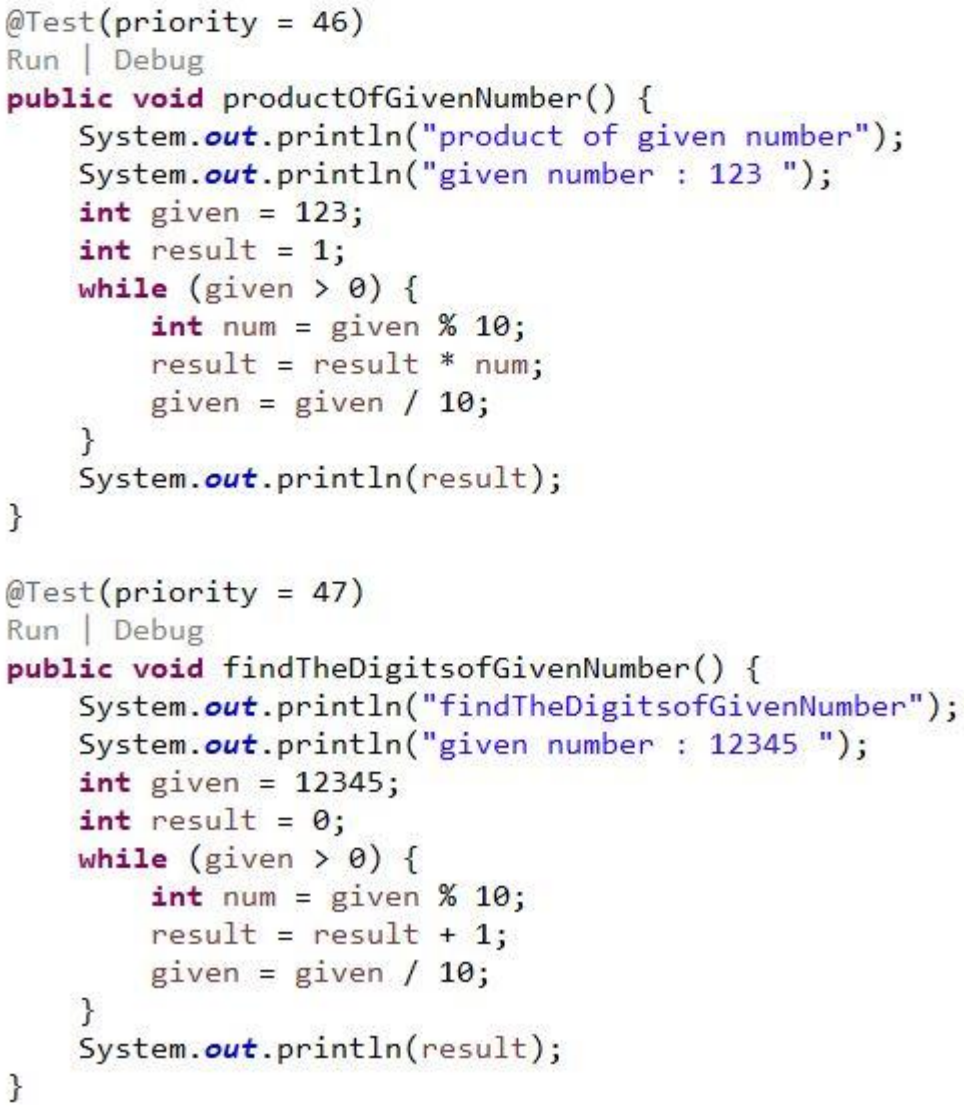
45. SUM OF THE GIVEN NUMBER



```java
@Test(priority = 44)
Run | Debug
public void reverseTheGivenNumber1() {
    System.out.println("reverseTheGivenNumber");
    System.out.println("givenNumber : 123 ");
    int given = 123;
    int result = 0;// to store the output
    while (given > 0) {
        int num = given % 10;
        result = (result * 10) + num;
        given = given / 10;
    }
    System.out.println(result);
}

@Test(priority = 45)
Run | Debug
public void sumOfGivenNumber() {
    System.out.println("sum of given number");
    System.out.println("given number : 123 ");
    int given = 123;
    int result = 0;
    while (given > 0) {
        int num = given % 10;
        result = result + num;
        given = given / 10;
    }
    System.out.println(result);
}
```

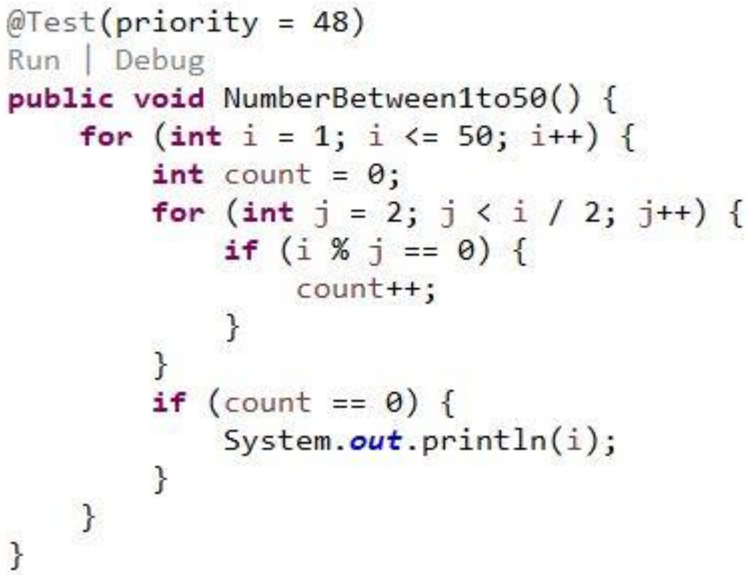
46. PRODUCT OF THE GIVEN NUMBER

47. FIND THE DIGITS OF THE GIVEN NUMBER



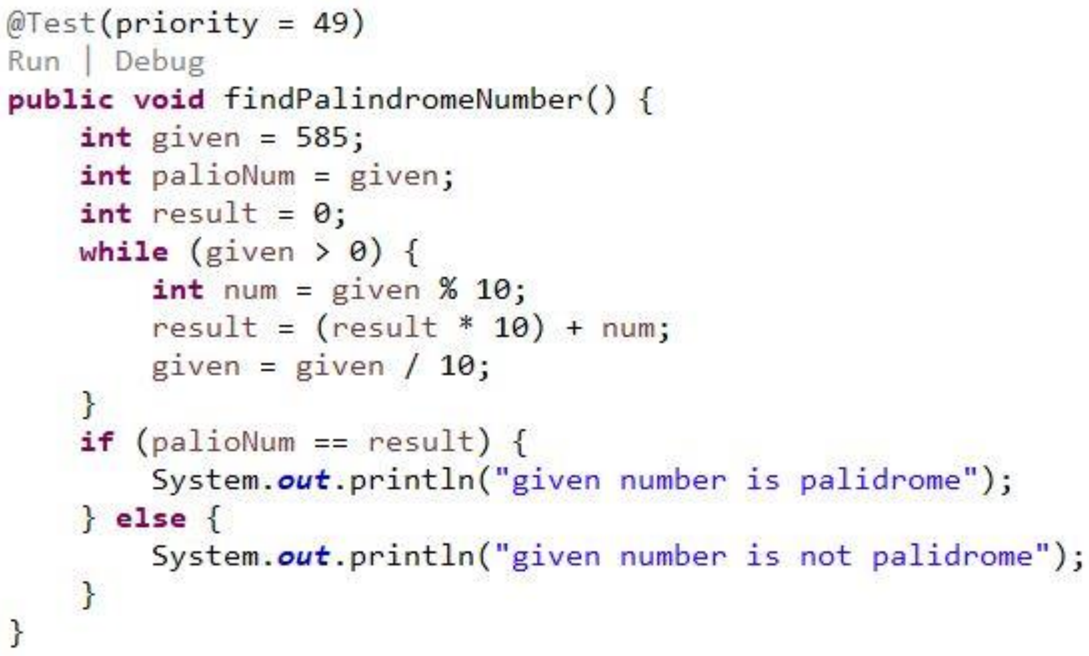
```
@Test(priority = 46)
Run | Debug
public void productOfGivenNumber() {
    System.out.println("product of given number");
    System.out.println("given number : 123 ");
    int given = 123;
    int result = 1;
    while (given > 0) {
        int num = given % 10;
        result = result * num;
        given = given / 10;
    }
    System.out.println(result);
}

@Test(priority = 47)
Run | Debug
public void findTheDigitsofGivenNumber() {
    System.out.println("findTheDigitsofGivenNumber");
    System.out.println("given number : 12345 ");
    int given = 12345;
    int result = 0;
    while (given > 0) {
        int num = given % 10;
        result = result + 1;
        given = given / 10;
    }
    System.out.println(result);
}
```

48. NUMBERS BETWEEN 1 TO 50



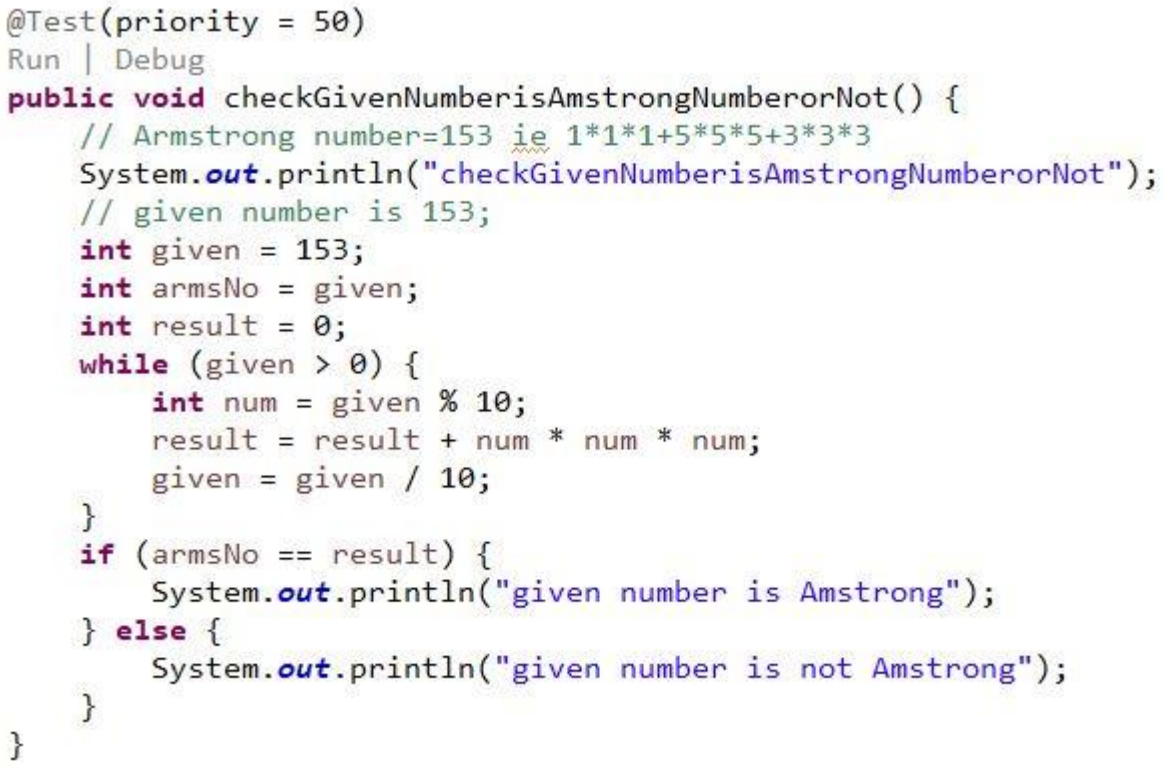
```
@Test(priority = 48)
Run | Debug
public void NumberBetween1to50() {
    for (int i = 1; i <= 50; i++) {
        int count = 0;
        for (int j = 2; j < i / 2; j++) {
            if (i % j == 0) {
                count++;
            }
        }
        if (count == 0) {
            System.out.println(i);
        }
    }
}
```

49. FIND THE PALINDROME NUMBER



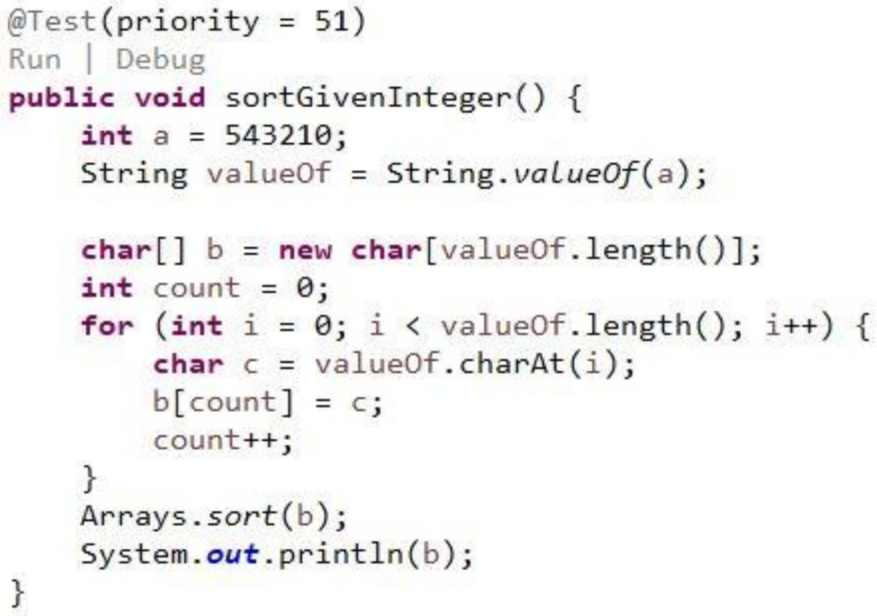
```
@Test(priority = 49)
Run | Debug
public void findPalindromeNumber() {
    int given = 585;
    int palioNum = given;
    int result = 0;
    while (given > 0) {
        int num = given % 10;
        result = (result * 10) + num;
        given = given / 10;
    }
    if (palioNum == result) {
        System.out.println("given number is palidrome");
    } else {
        System.out.println("given number is not palidrome");
    }
}
```

50. GIVEN NUMBER IS ARMSTRONG OR NOT



```
@Test(priority = 50)
Run | Debug
public void checkGivenNumberisAmstrongNumberorNot() {
    // Armstrong number=153 ie 1*1*1+5*5*5+3*3*3
    System.out.println("checkGivenNumberisAmstrongNumberorNot");
    // given number is 153;
    int given = 153;
    int armsNo = given;
    int result = 0;
    while (given > 0) {
        int num = given % 10;
        result = result + num * num * num;
        given = given / 10;
    }
    if (armsNo == result) {
        System.out.println("given number is Amstrong");
    } else {
        System.out.println("given number is not Amstrong");
    }
}
```

51. SORT THE GIVEN INTEGER



```
@Test(priority = 51)
Run | Debug
public void sortGivenInteger() {
    int a = 543210;
    String valueOf = String.valueOf(a);

    char[] b = new char[valueOf.length()];
    int count = 0;
    for (int i = 0; i < valueOf.length(); i++) {
        char c = valueOf.charAt(i);
        b[count] = c;
        count++;
    }
    Arrays.sort(b);
    System.out.println(b);
}
```