

```

public class ArrayOps {
    public static void main(String[] args) {

        int[] testCase1 = {2,8,3,7,8};

        System.out.println(secondMaxValue(testCase1));
    }

    public static int findMissingInt (int[] array) {
        int n = array.length;

        for (int i = 0; i <= n; i++) {
            boolean number = false;
            for (int index = 0; index < array.length; index++) {
                if (array[index] == i) {
                    number = true;
                }
            }
            if (number == false) {
                return i;
            }
        }
        return 1;
    }

    public static int secondMaxValue(int [] array) {
        int l = array.length;
        int large = 0;
        int secondLarger = 0;

        for (int i = 0; i < l; ++i) {
            if (array[i] > secondLarger) {
                secondLarger = large;
                large = array[i];
            }
        }
        return secondLarger;
    }

    public static boolean containsTheSameElements(int [] array1,int [] array2) {
        int n = array1.length;
        int m = array2.length;
    }
}

```

```

for (int i = 1; i < n; i++) {
    for (int j = 0; j < m; j++) {
        if (array1[i] == array2[j]) {
            break;
        }
        else if (j == (m - 1)) {
            return false;
        }
    }
}

for (int i = 1; i < m; i++) {
    for (int j = 0; j < n; j++) {
        if (array2[i] == array1[j]) {
            break;
        }
        else if (j == (n - 1)) {
            return false;
        }
    }
}
return true;
}

```

```

public static boolean isSorted(int [] array) {
    // Write your code here:
    if (array.length <= 1) {
        return true;
    }

    boolean minOrMax = array[0] < array[1];

    for (int i = 1; i < array.length; i++) {
        if (minOrMax) {
            if (array[i - 1] > array[i]) {
                return false;
            }
        }
        else if (array[i - 1] < array[i]) {
            return false;
        }
    }
}

```

```
    return true;  
  }  
}
```

```

public class StringOps {
    public static void main(String[] args) {

    }

    private static char characterToLowerCase(char c) {
        if (c >= 'A' && c <= 'Z') {
            return (char)(c + 32);
        }
        return c;
    }

    private static char characterToUpperCase(char c) {
        if (c >= 'a' && c <= 'z') {
            return (char)(c - 32);
        }
        return c;
    }

    private static boolean characterIsVowel(char c) {
        switch (c) {
            case 'a':
            case 'e':
            case 'i':
            case 'o':
            case 'u':
            case 'A':
            case 'E':
            case 'I':
            case 'O':
            case 'U':
                return true;
            default:
                return false;
        }
    }

    public static String capVowelsLowRest (String string) {
        String result = "";

        for (int i = 0; i < string.length(); ++i) {
            if (characterIsVowel(string.charAt(i))) {
                result += characterToUpperCase(string.charAt(i));
            } else {

```

```

        result += characterToLowerCase(string.charAt(i));
    }
}
return result;
}

public static String camelCase (String string) {
    String result = "";

    for (int i = 0; i < string.length(); ++i) {
        while (string.charAt(i) == ' ') {
            ++i;
        }

        // we found first character already, make first character
        // of new word uppercase
        if (result != "" && string.charAt(i - 1) == ' ') {
            result += characterToUpperCase(string.charAt(i));
        } else {
            result += characterToLowerCase(string.charAt(i));
        }
    }
    return result;
}

public static int[] allIndexOf (String string, char chr) {
    int indexOfCharacter = -1;
    int indexesFound = 0;

    do {
        indexOfCharacter = string.indexOf(chr, indexOfCharacter + 1);
        if (indexOfCharacter != -1) {
            indexesFound++;
        }
    } while (indexOfCharacter != -1);

    int[] result = new int[indexesFound];
    int indexIndex = 0;

    do {
        indexOfCharacter = string.indexOf(chr, indexOfCharacter + 1);
        if (indexOfCharacter != -1) {
            result[indexIndex++] = indexOfCharacter;
        }
    }

```

```
    } while (indexOfCharacter != -1);  
    return result;  
  }  
}
```