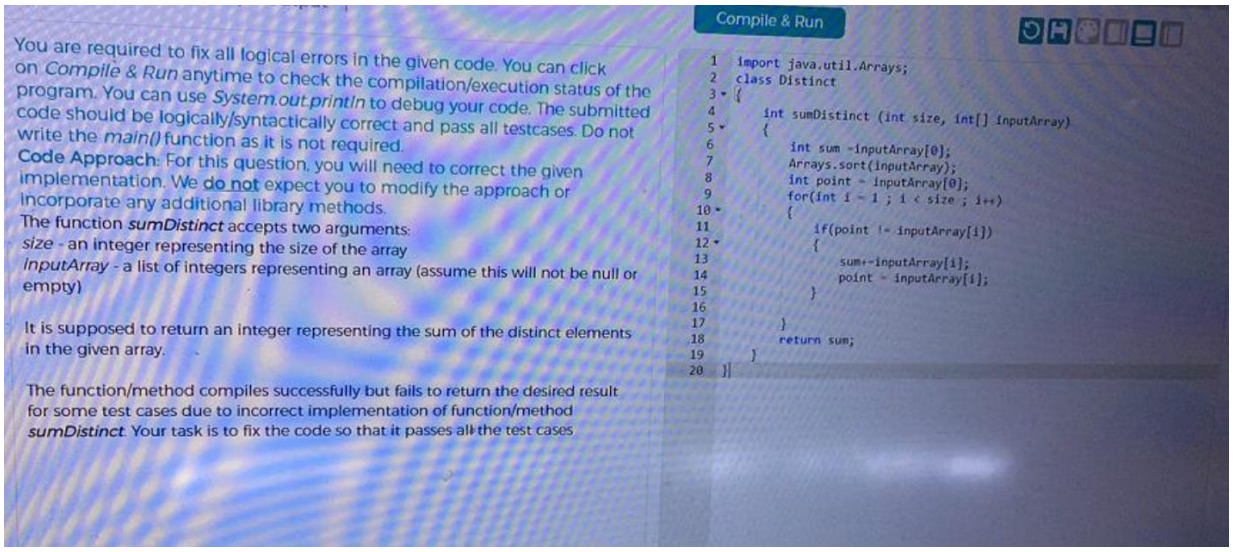
1. **distinctNum**

   == 改为 !=

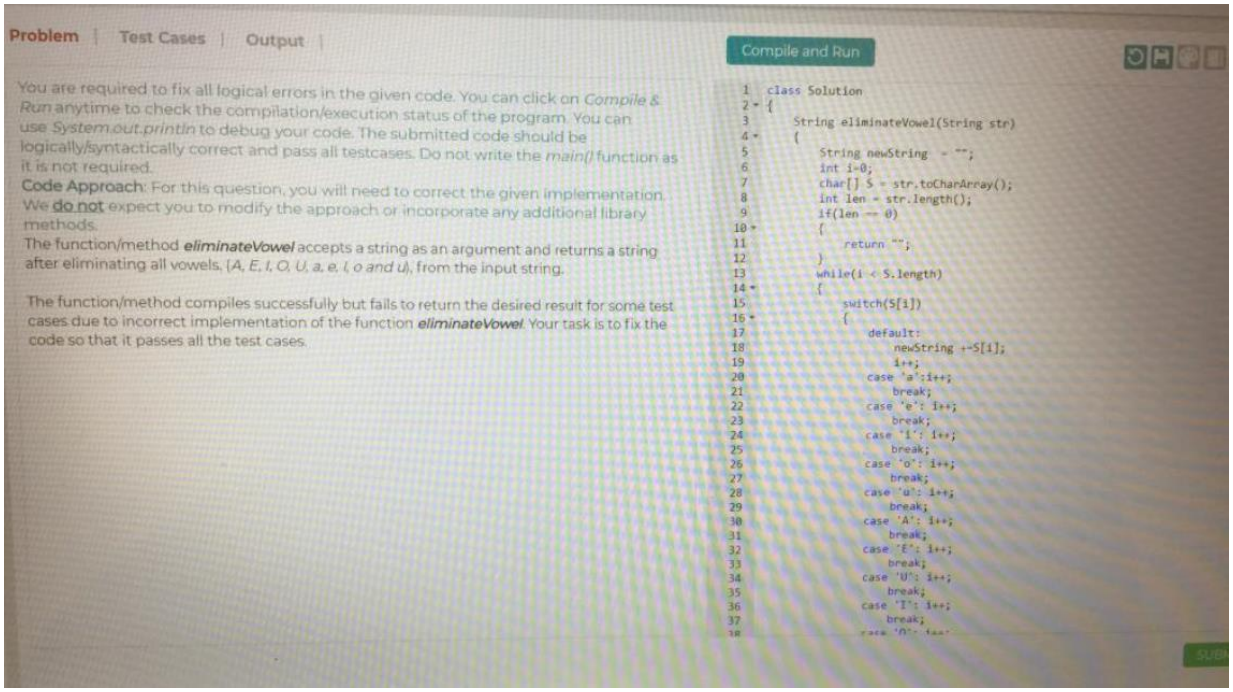
2. **sumDistinct**

   初始化sum数组写在sort之后



3. **eliminateVowel**

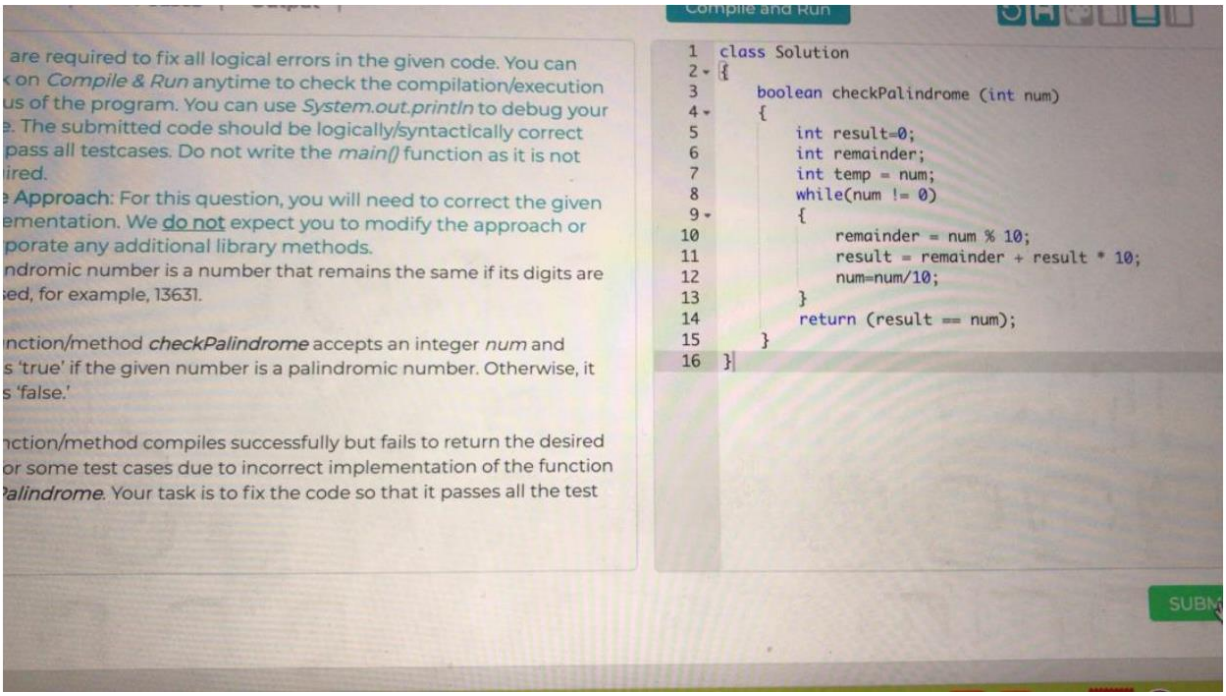
   default 去掉 i++



4. **checkPalindrome**

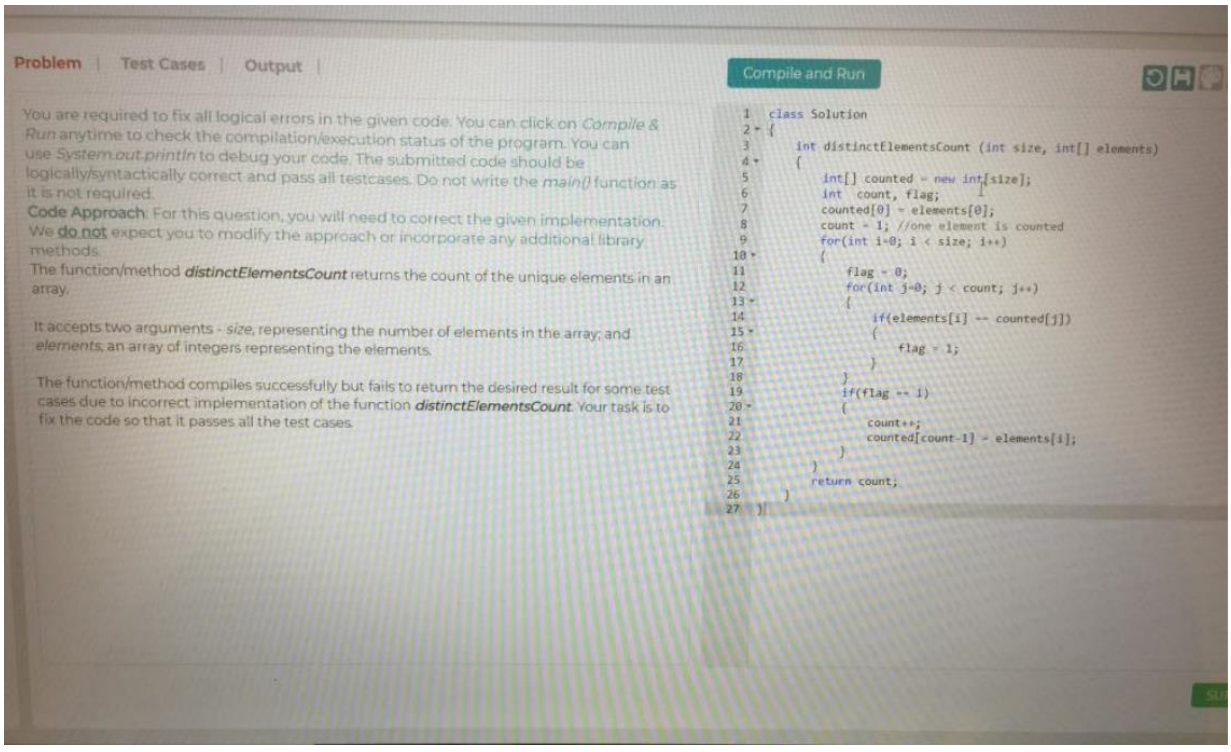
   把最后返回值改成 return (result == temp);

   result = result * 10 + remainder ????

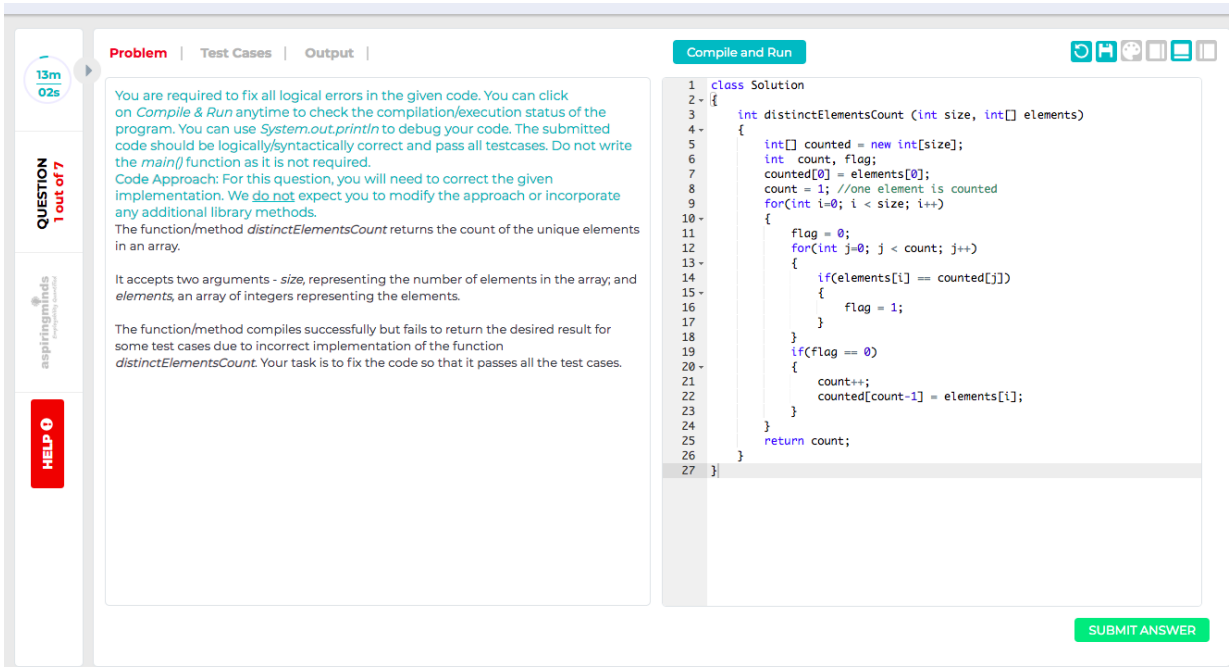


5. distinctElementCount

if(flag == 1) 改成 if(flag == 0)



下面是改后



6. matrixSum

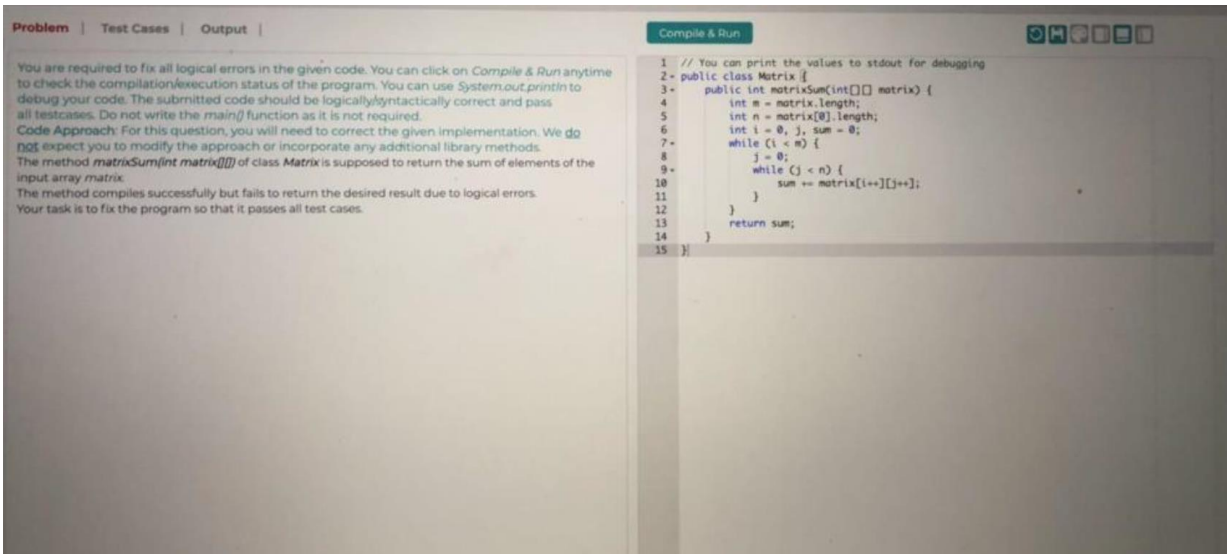
while(j < n)

sum += matrix[i++][j++]

改成

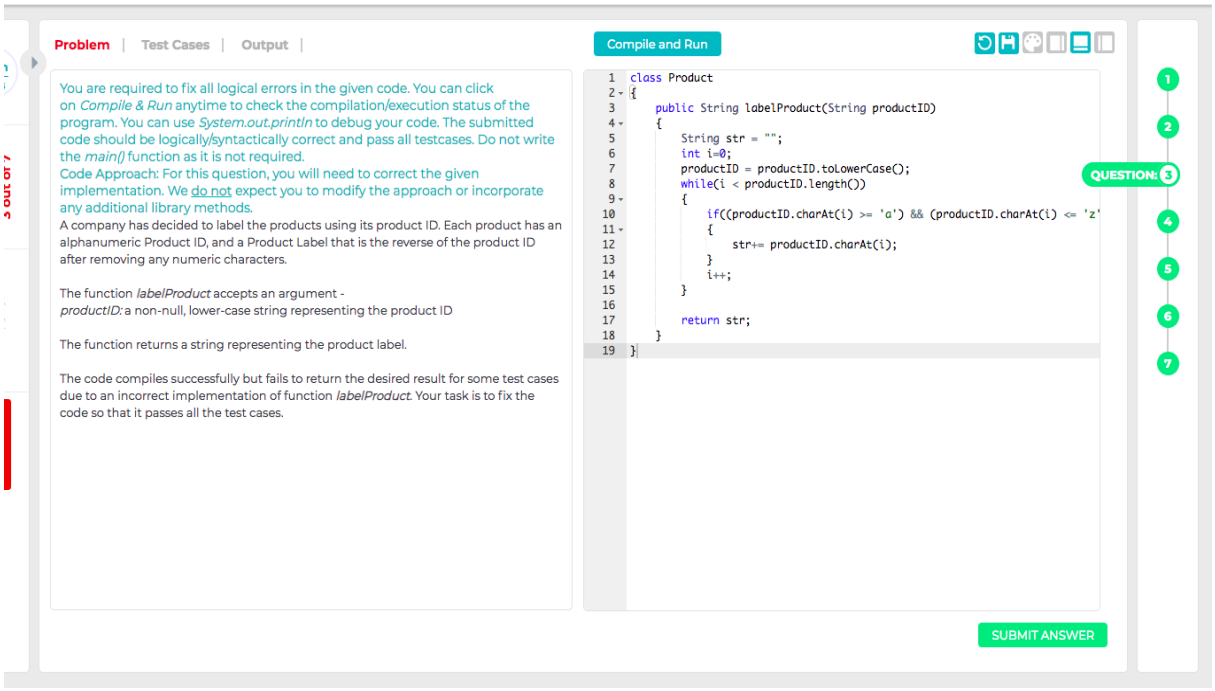
while(j < n) { sum += matrix[i][j++]} i++;

或者 i++ 和 j++换位置？

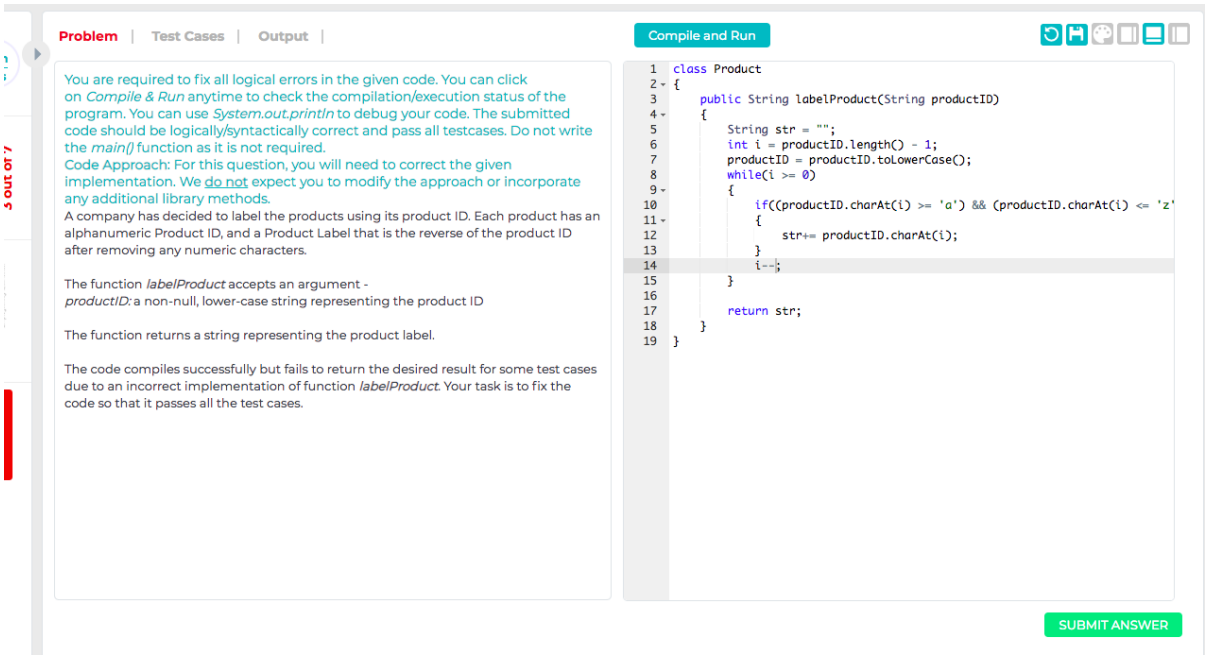


7. labelProduct

加入入字符串串的方方向反了了 while 循环中



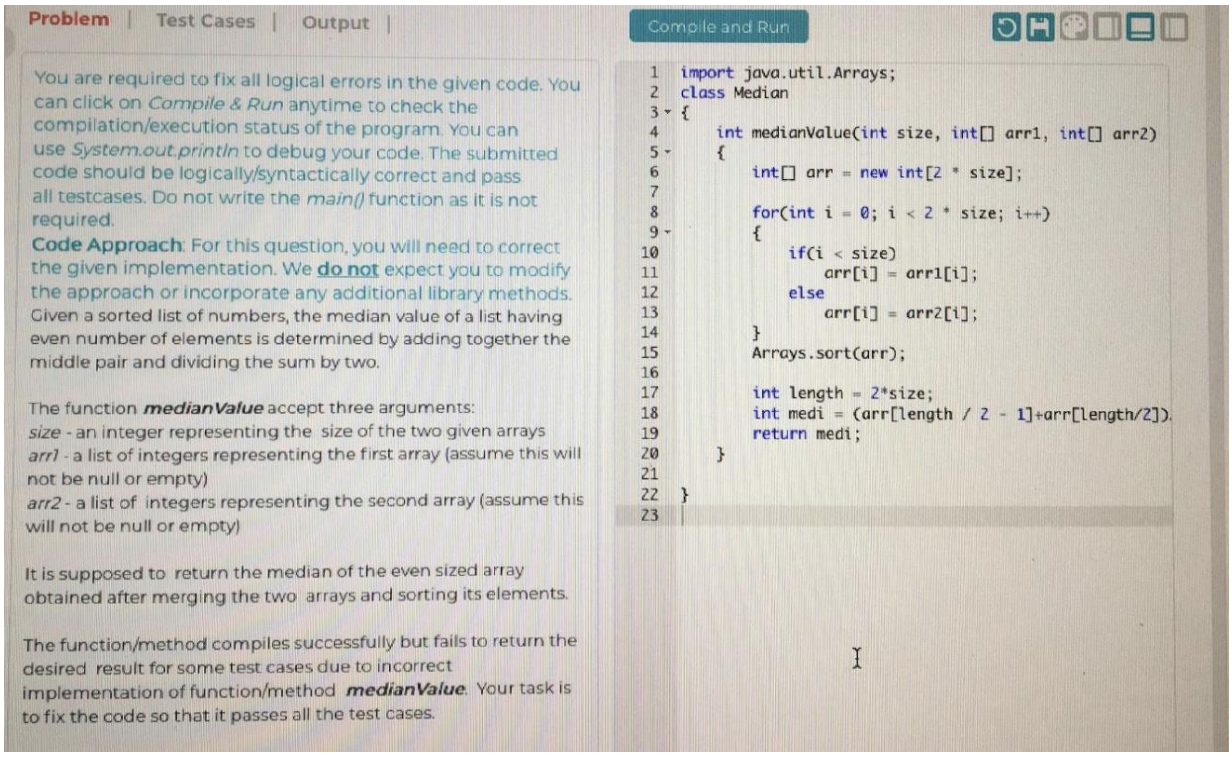
改之后



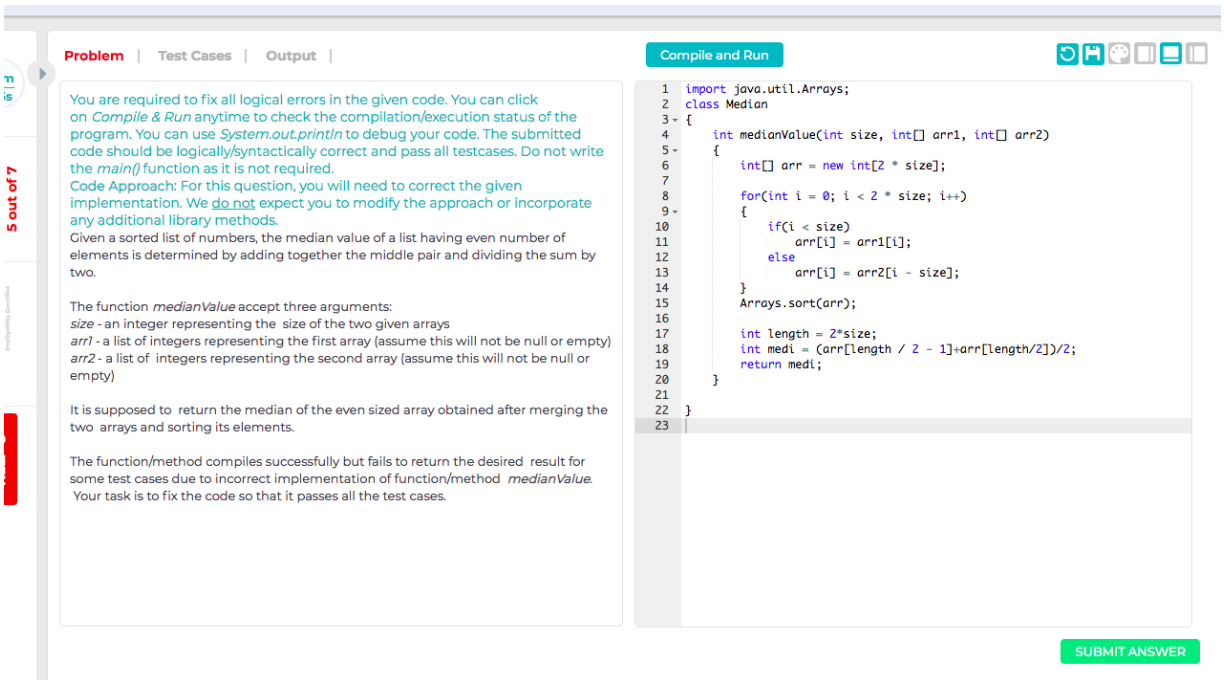
8. medianValue

arr2 越界，改为 arr2[i-size]

else arr[i] = arr2[i] → else arr[i] = arr2[i - size]



改之后

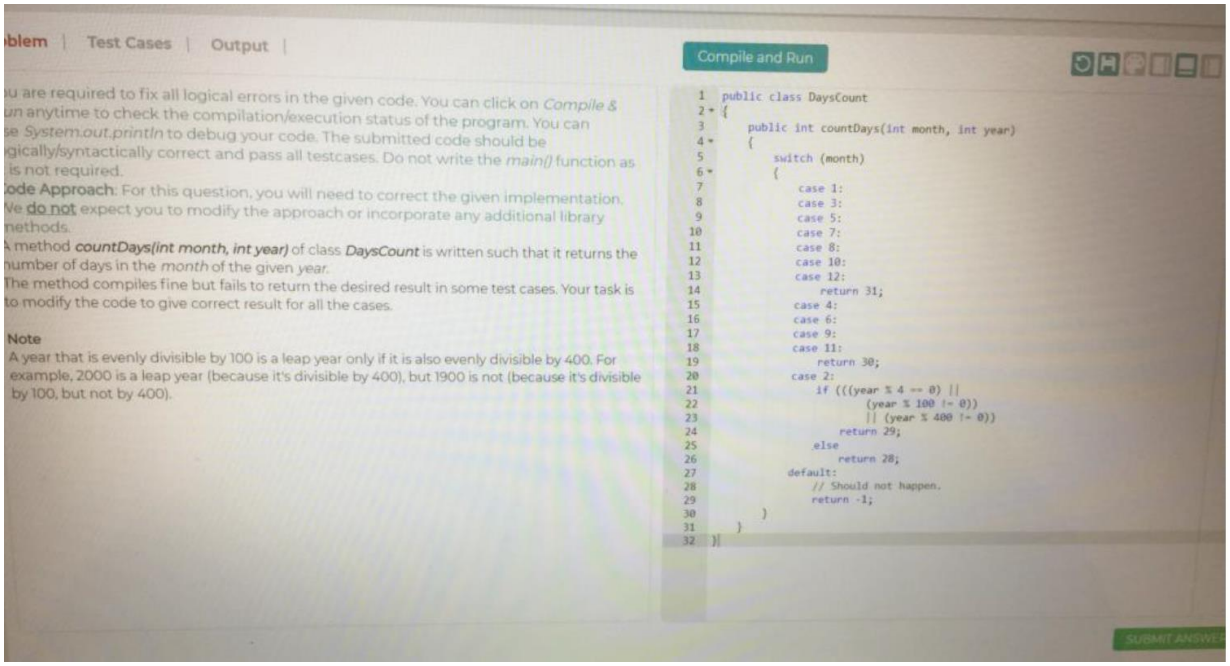


9. **countDays**

(year % 4 == 0) && (year %100 != 0) || (year % 400 == 0)

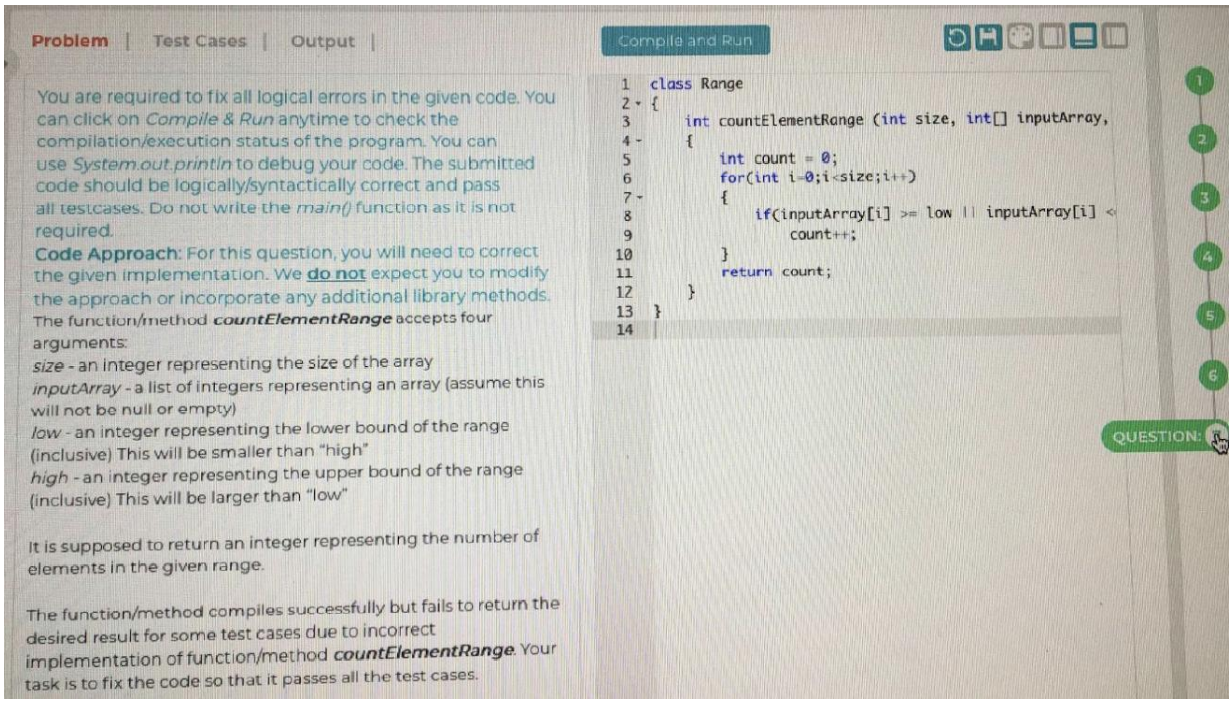
((y % 4) == 0 || (y % 100) != 0) 应改为 ((y % 4) == 0 && (y % 100) != 0)

(((year % 400 == 0 && year % 3200 != 0) || (year % 100 != 0)) && (year % 4 == 0))



10. countElementRange

把 || 改成 &&

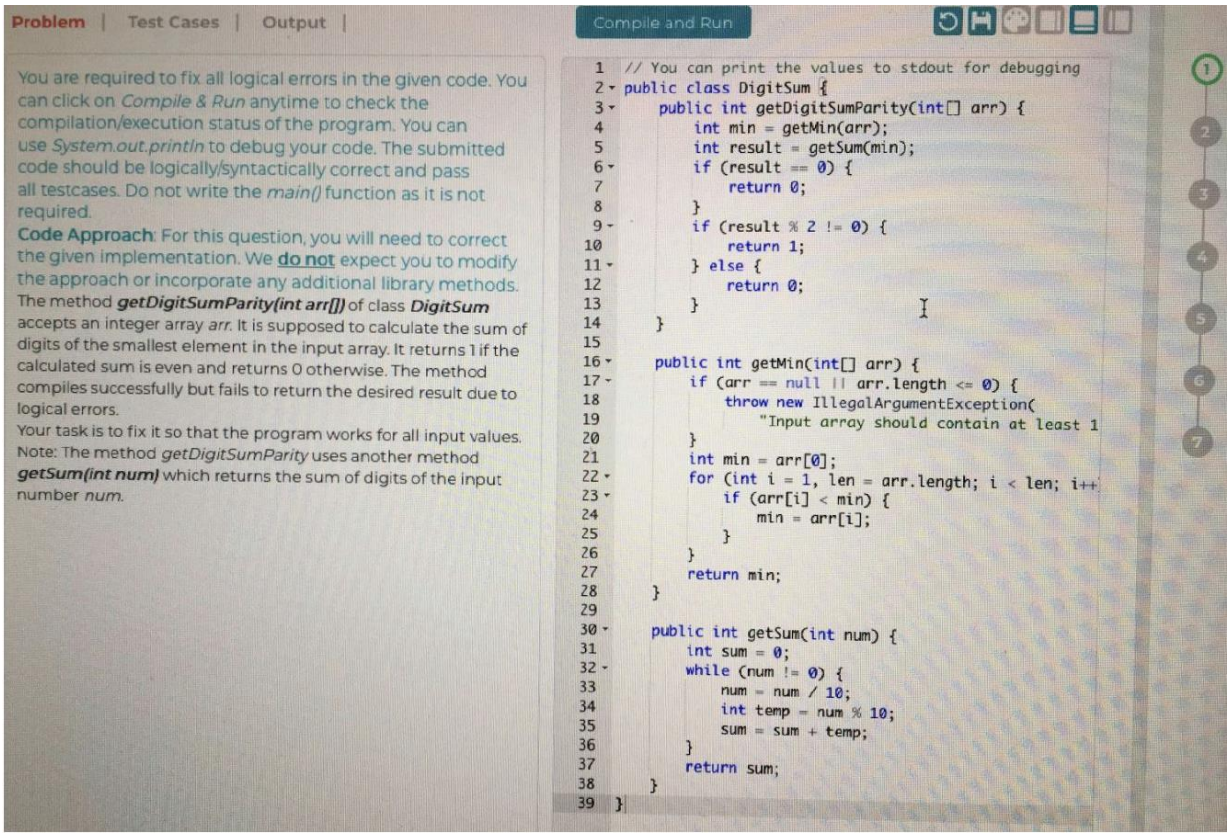


11. countNumParity getDigitSumParity DigitSum

(1) 主函数 result % 2 == 0

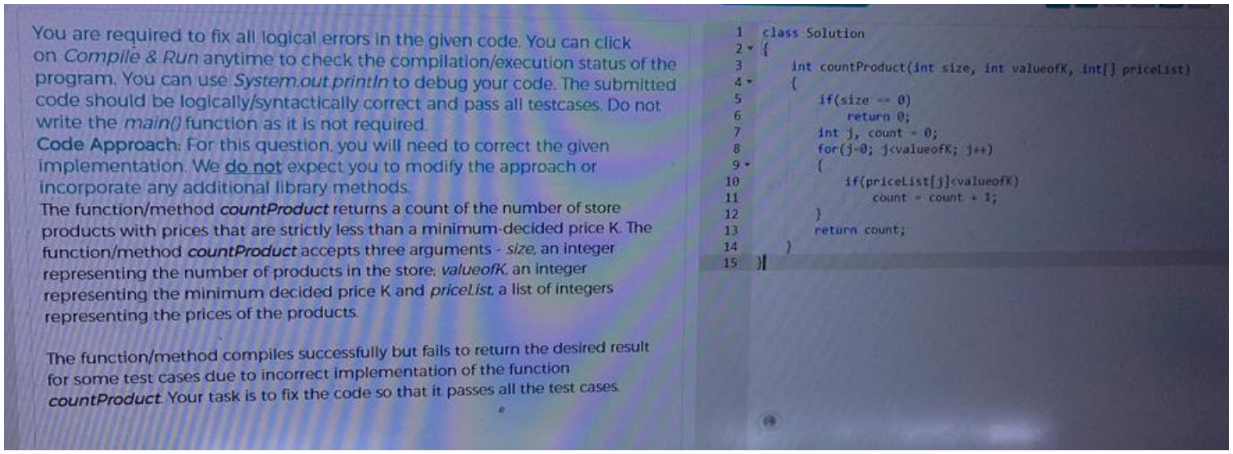
(2) 主函数 if (result == 0) {return 1}

(3) sumDigit 函数把 int temp = num % 10 和 num = num/10 换一一下位置

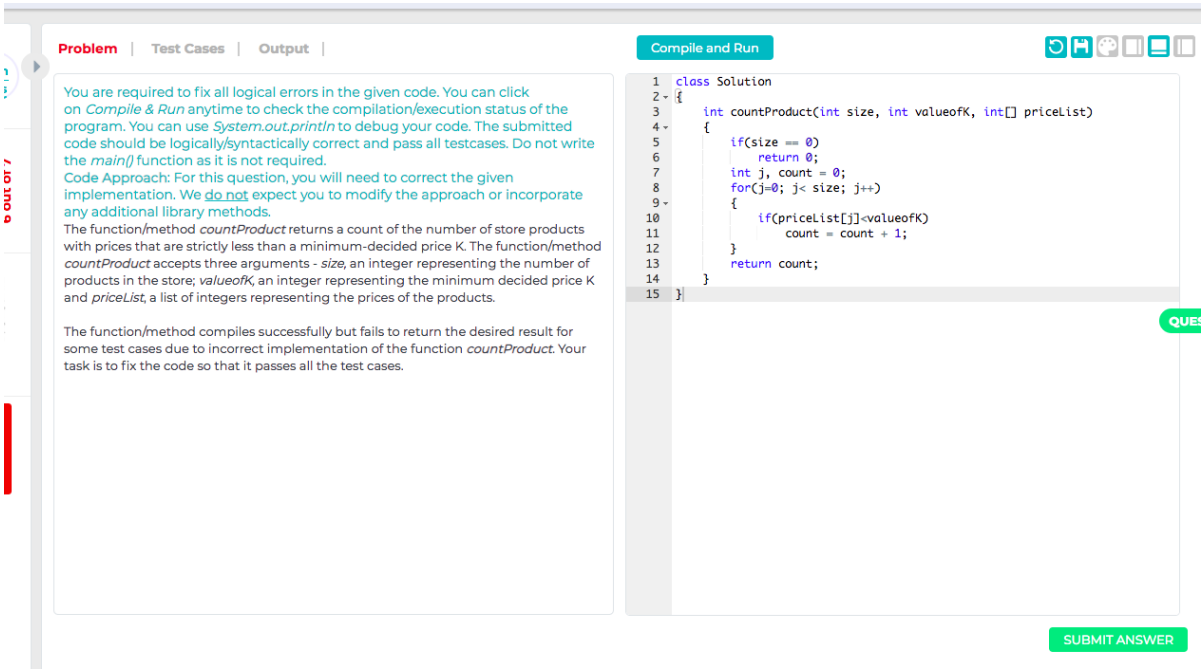


12. countProduct

for (j = 0; j< value; j++) 改为 for (j = 0; j < size; j++);



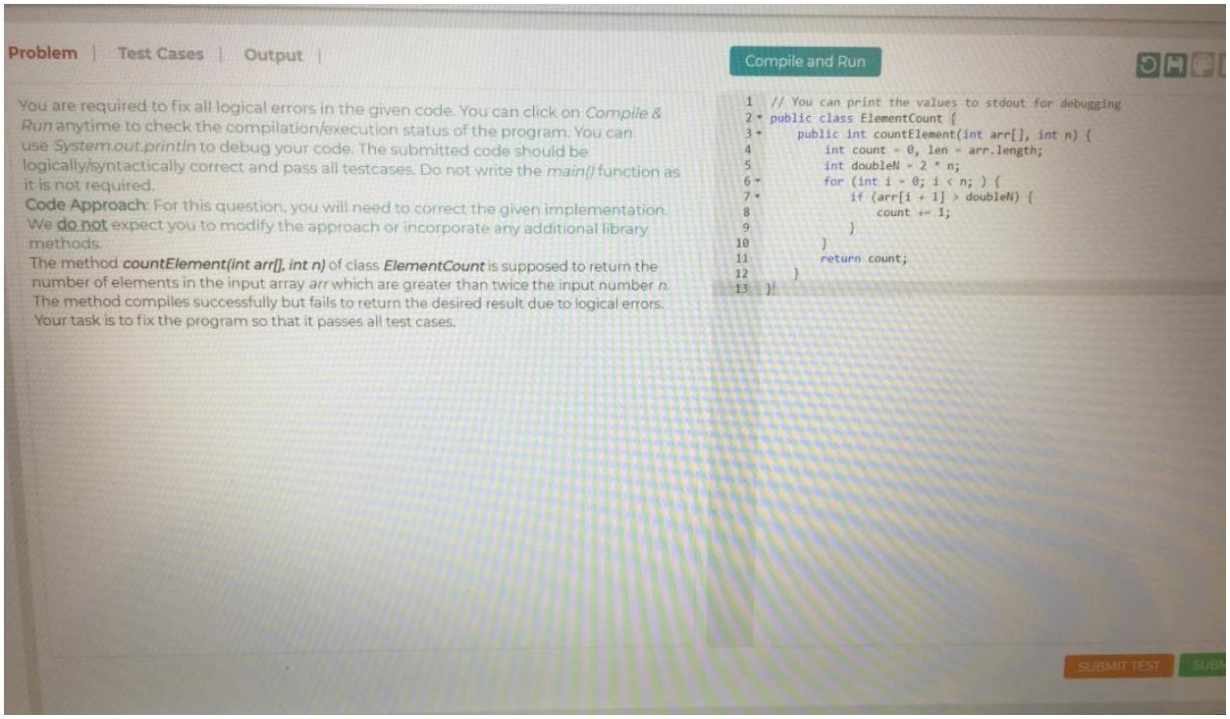
改之后



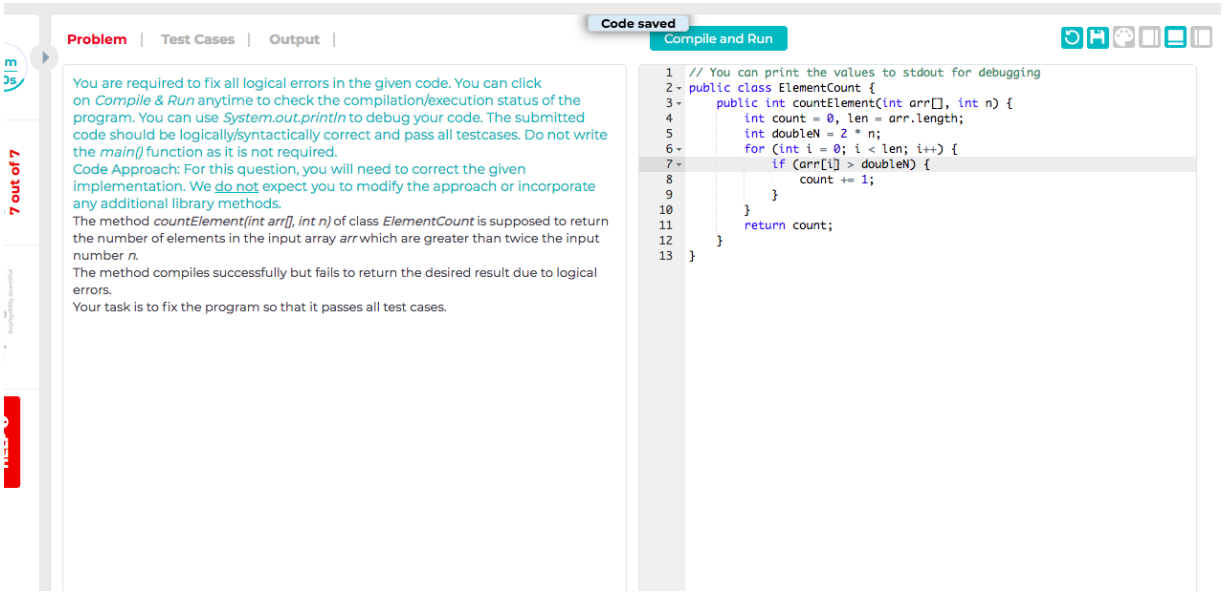
13. countElement

改成 i < len; i++

后面面 arr[i+1] 改成 arr[i]



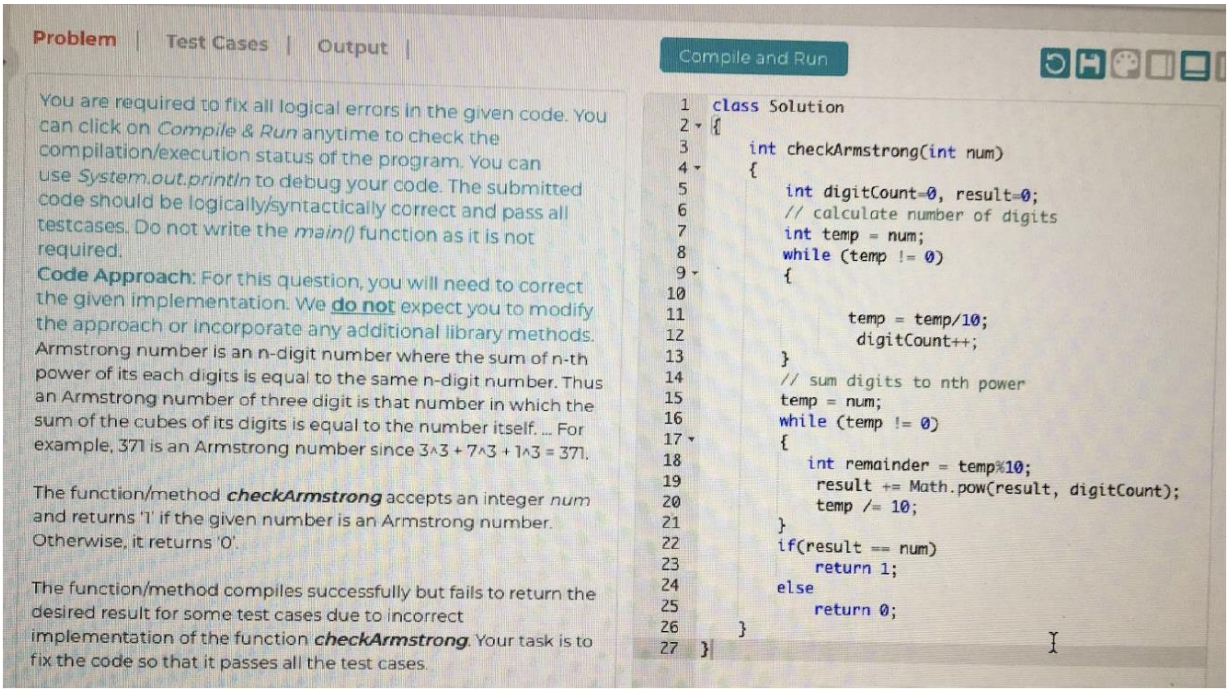
下面是改后



14. checkArmstrong

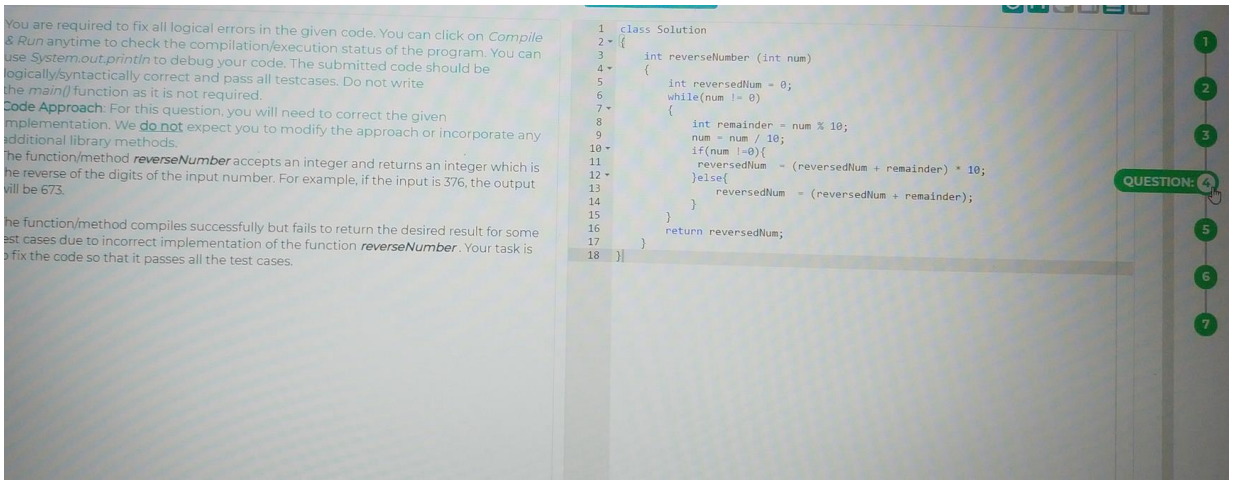
result+=math.pow(result, digitCount)应该改为

result+=math.pow(remainder, digitCount)



15. reverseNumber

应该是 reverseNum = reverseNum*10 + Reminder



16. appearsKTimes

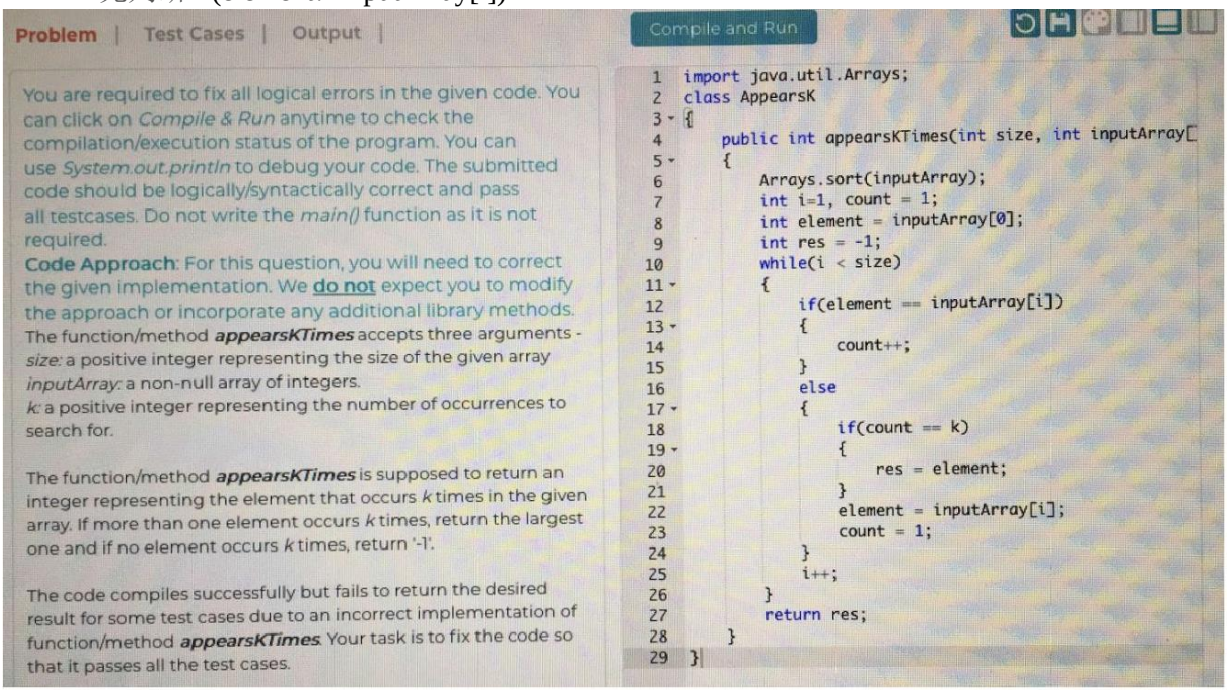
while loop 结束后 添加

if (count == k) { res = element }

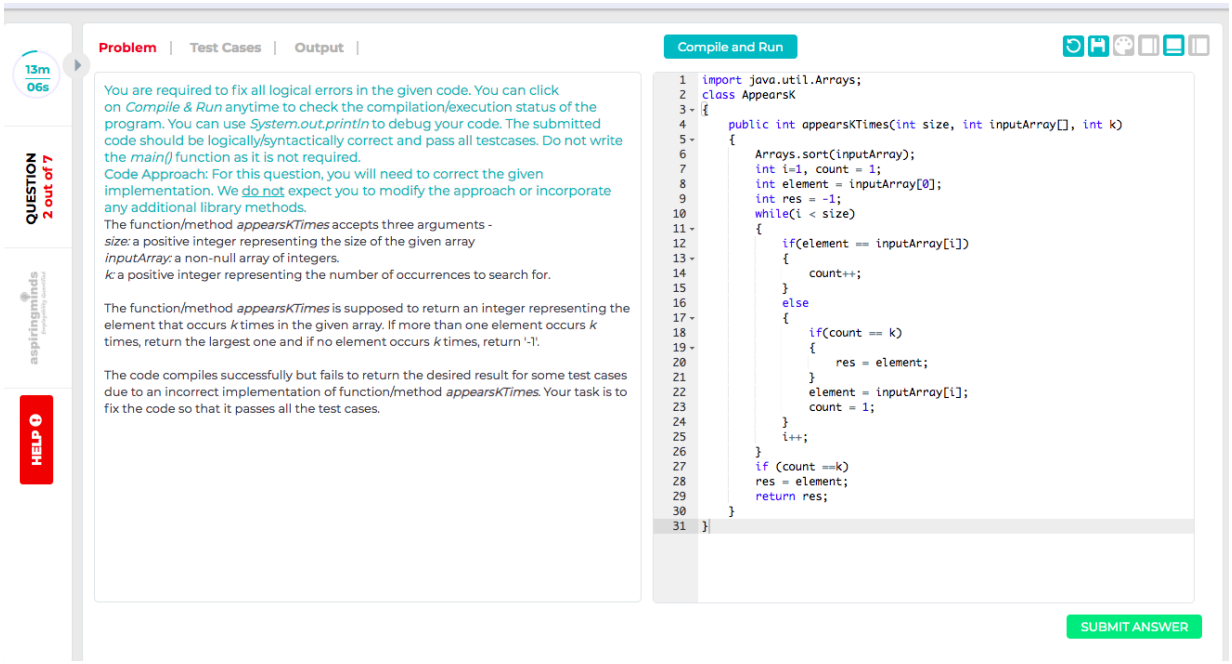
if 和 else 内容 换一一下位置,

先判断 if(element!=inputArray[i])

while loop 后加个check, 并且 res = Math.max(res, element)



下面是改后



17. reverseString

去掉 len++?

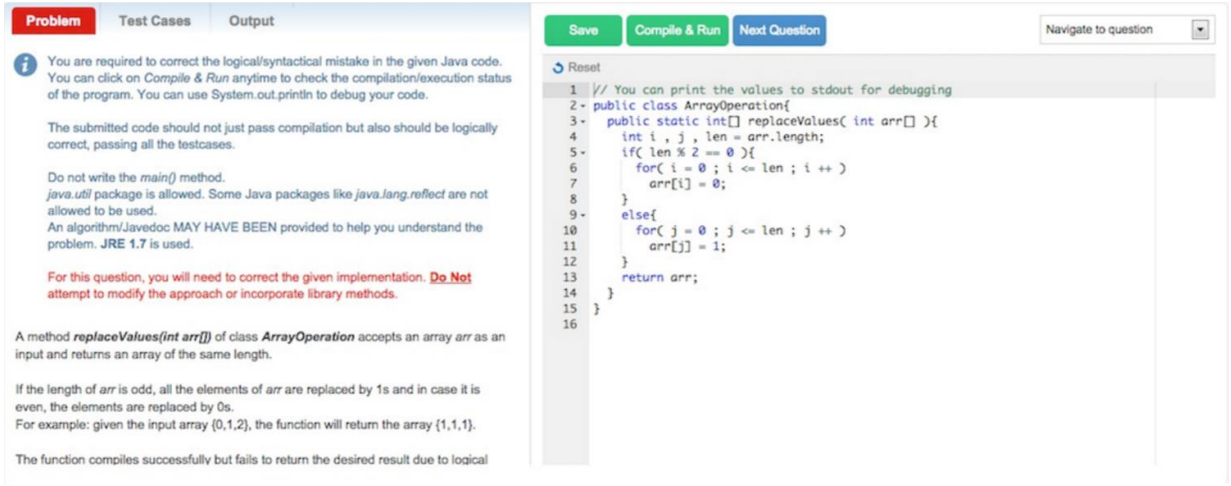
不不是 str += ？ ？ ？

18. replaceValues

若 array 长度为奇数全换为 1 偶数换为 0

或者

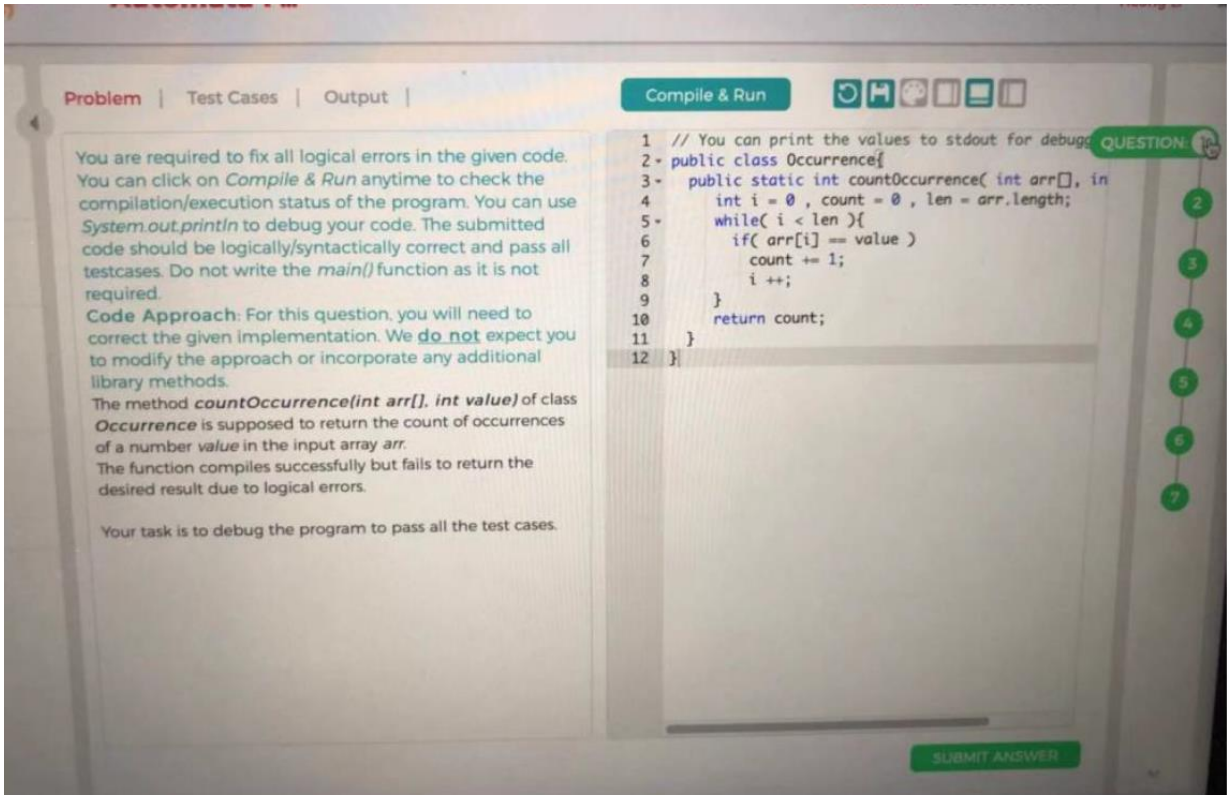
for loop 中 i<=len; j<=len 改成 i<len, j<len



```java
/**
 * Created by yaqunyu on 10/20/16.
 */
public class ArrayOperation {
    public static int[] replaceValues(int arr[]) {
        int i, j, len = arr.length;
        if(len % 2 == 0){
            for(i = 0; i<len; i++) {
                arr[i] = 0;
            }
        }
        else{
            for(j = 0; j<len; j++){
                arr[j] = 1;
            }
        }
        return arr;
    }
    public static void main(String[] args){
        int[] myIntArray = new int[]{20,2,3,5,9,6,16};
        for(int i = 0; i<myIntArray.length; i++) {
            System.out.println(replaceValues(myIntArray)[i]);
        }
    }
}
```

19. countOccurences

返回 value 在 array 中出现的次数, while 里面最后加上 i++（error: TLE）

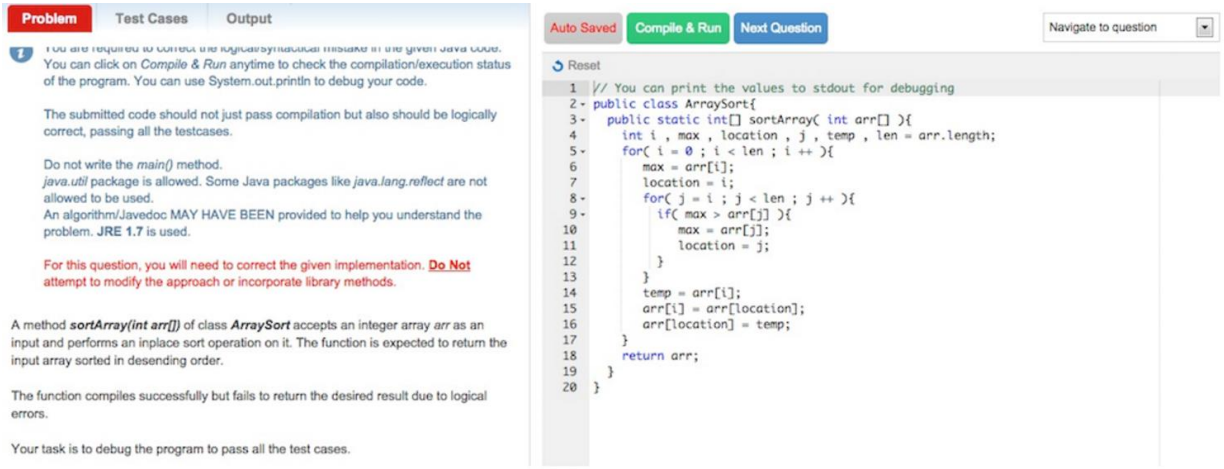


20. checkGrade gradingsystem

成绩打分 ABCD，判断输入的数在什么范围, 它上面是 （x >= 70) || (x < 80), 两个|| 改成 &&

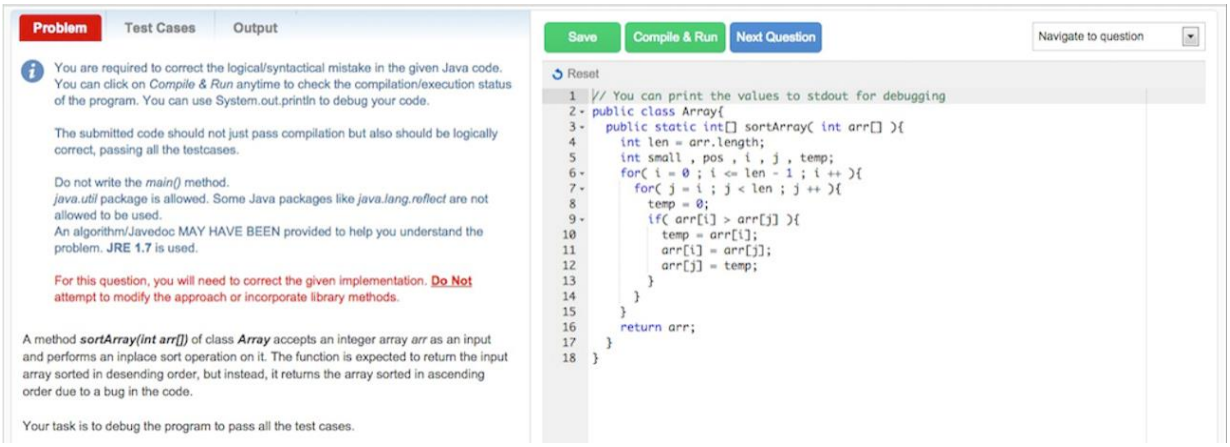
21. sortArray descendingSortArray

if (max > arr[j]) 改成 <



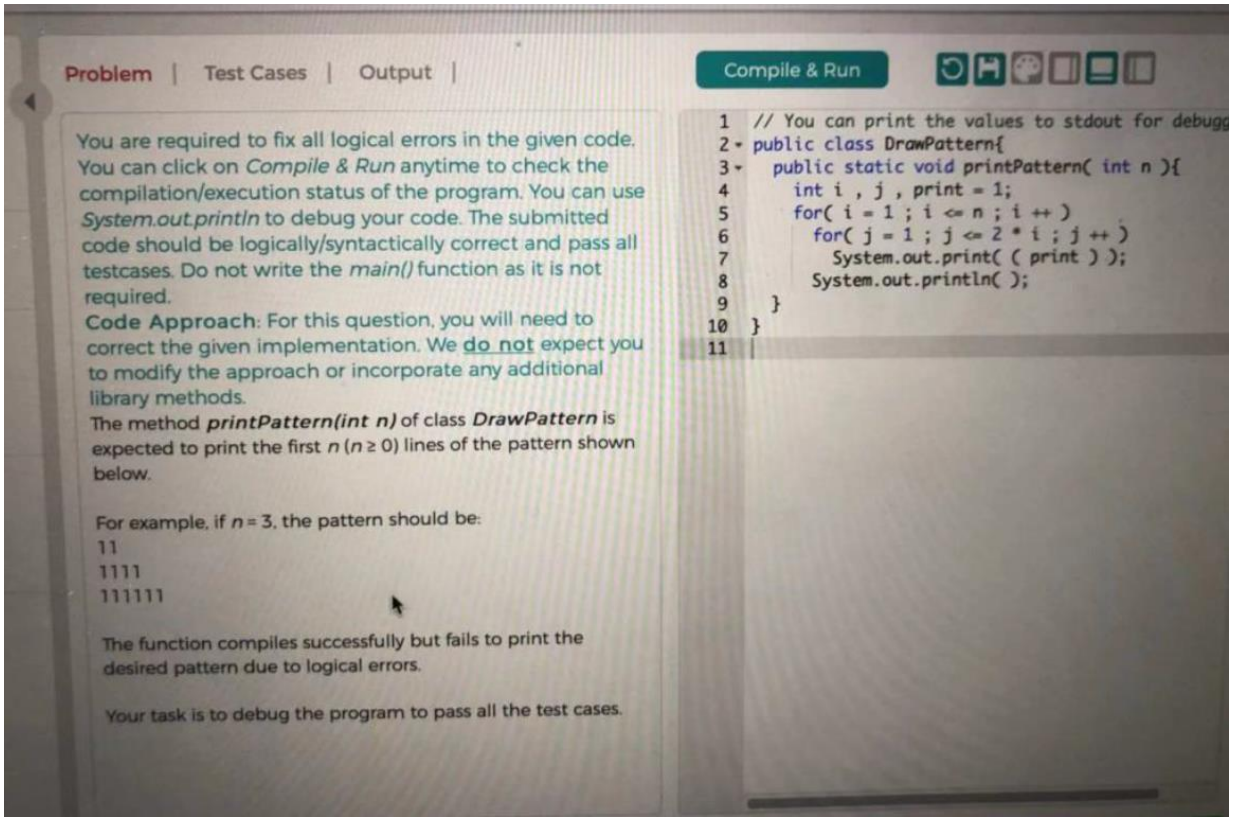
Sort Array

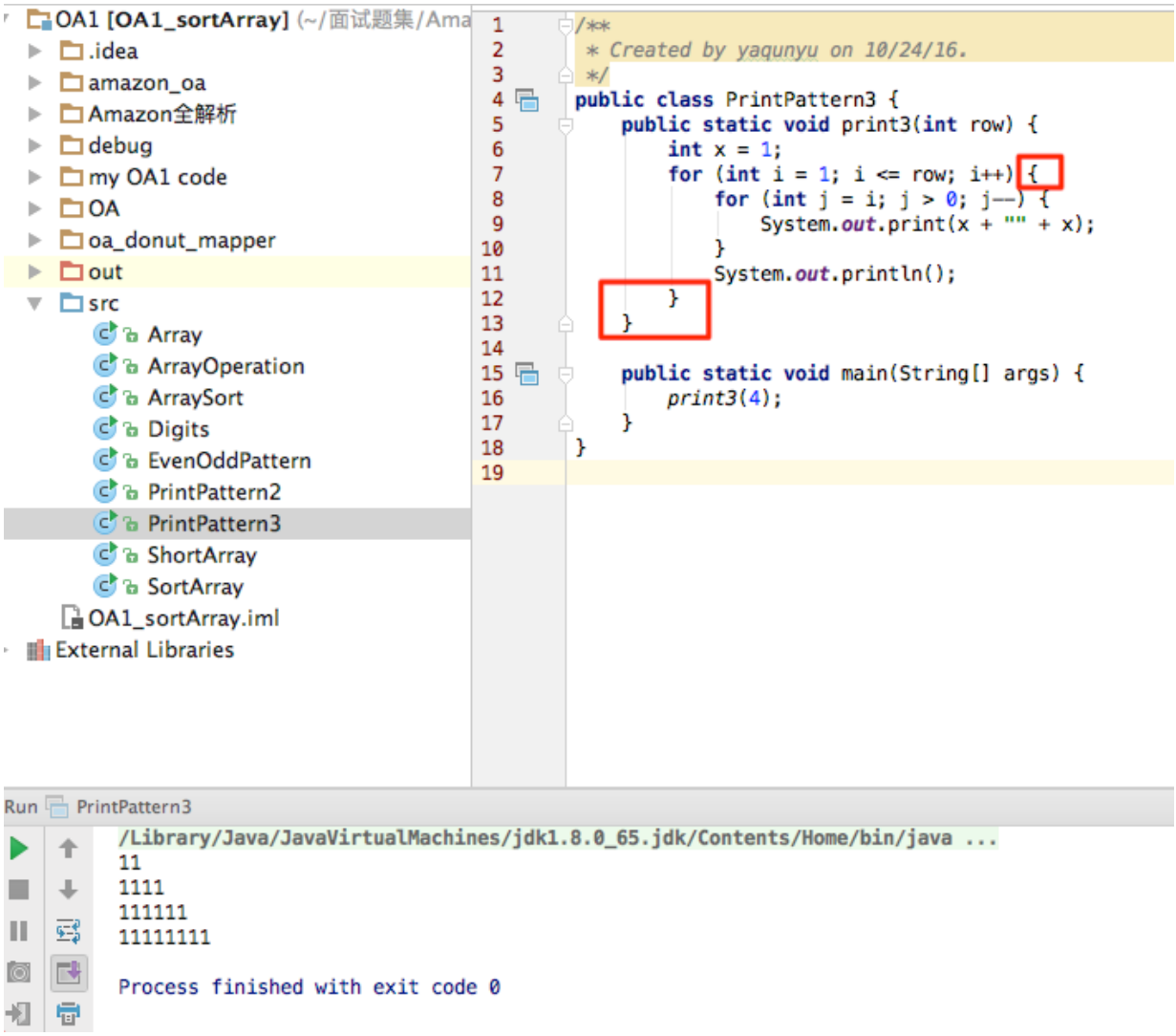
if (arr[i] > arr[j]) 改成 <



22. printPattern

第一种情况　给 for loop 加括号{}

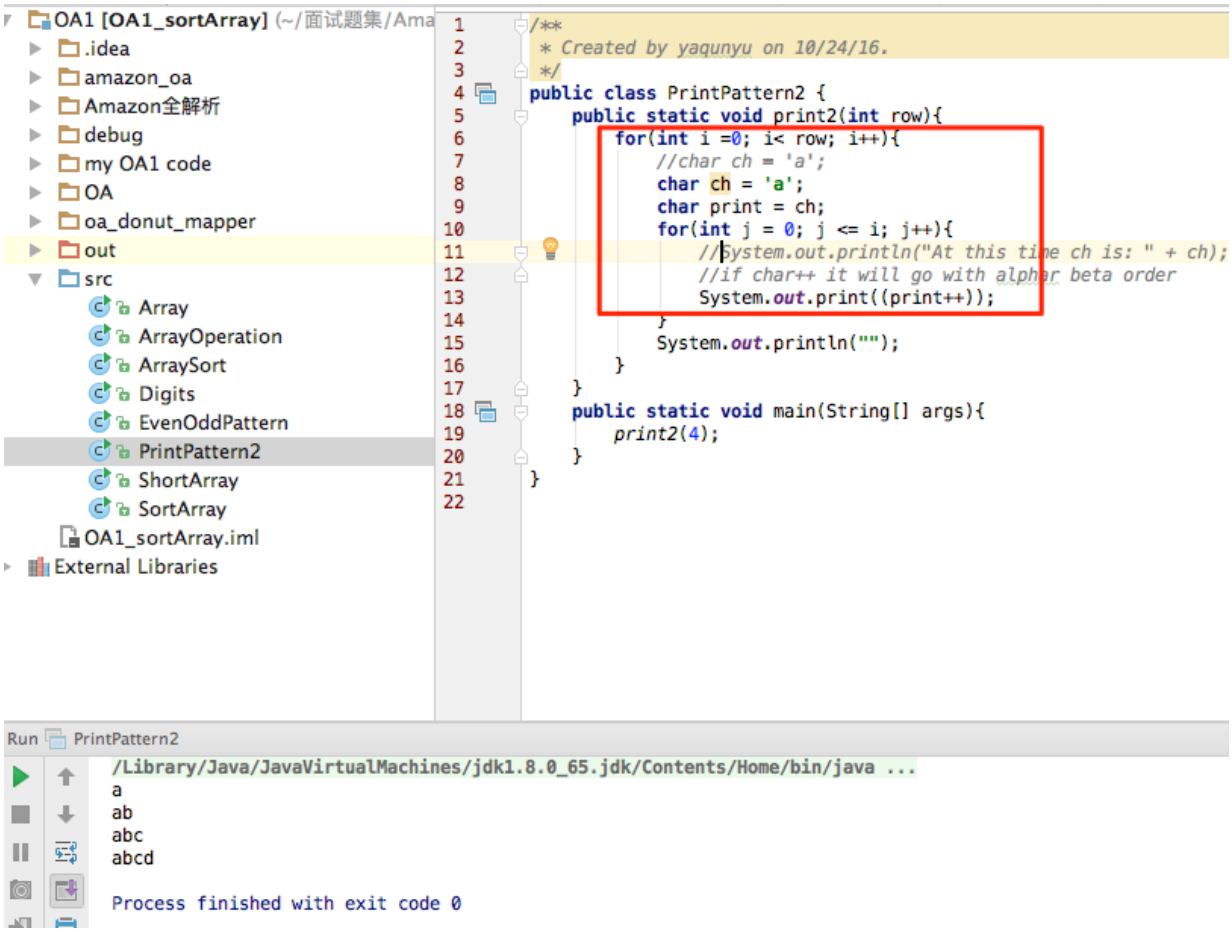

Problem | Test Cases | Output |
Compile & Run
You are required to fix all logical errors in the given code. You can click on Compile & Run anytime to check the compilation/execution status of the program. You can use System.out.println to debug your code. The submitted code should be logically/syntactically correct and pass all testcases. Do not write the main() function as it is not required.
Code Approach: For this question, you will need to correct the given implementation. We do not expect you to modify the approach or incorporate any additional library methods.
The method printPattern(int n) of class DrawPattern is expected to print the first n (n ≥ 0) lines of the pattern shown below.
For example, if n = 3, the pattern should be:
11
1111
111111
The function compiles successfully but fails to print the desired pattern due to logical errors.
Your task is to debug the program to pass all the test cases.
1 // You can print the values to stdout for debugg
2 public class DrawPattern{
3 public static void printPattern(int n){
4 int i , j , print = 1;
5 for(i = 1 ; i <= n ; i ++)
6 for(j = 1 ; j <= 2 * i ; j ++)
7 System.out.print((print));
8 System.out.println( );
9 }
10 }
11



第二种情况　输出 a ab abc abcd cout(ch++) 改为 cout(print++),

改为

```
char ch = 'a';
char print = ch;
for(int j = 0; j <=i; j++) {
    System.out.print((print++));
}
System.out.println("");
```



23. selectionSort　selectionSortArray

ascending order, arr[index_of_min] > arr[x] 改为 arr[y]

distinctNum: 把==改成!=

sunDistince:初始化 sum 数组写在 sort 之后

```java
import java.util.Arrays;
class Distinct
{
    int sumDistinct (int size, int[] inputArray)
    {
        Arrays.sort(inputArray);
        int sum =inputArray[0];

        int point = inputArray[0];
        for(int i = 1 ; i < size ; i++)
        {
            if(point != inputArray[i])
            {
                sum+=inputArray[i];
                point = inputArray[i];
            }

        }
        return sum;
    }
}
```

eliminatedVowel: default condition 后面的 i++删掉

checkPanlidrome: 把 result == num 改成 result == temp

DistinctElementCount: 感觉好像和之前看到的题库有些不同？==改了!=还是不对。

if (flag == 1) → if (flag == 0)

MatrixSum: i++和 j++位置

LabelProduct: 加入字符串的方向反了

CheckPalindrome: return 的 num 改成 temp

SumArray: 初始化 sum 数组写在 sort 之后

median: arr2 *改成 arr2[i-size]*

countDays:闰年 (year %4 == 0) && (year %100 != 0) || (year%400 == 0)

elementRange: 把|| 改成&&

countNumParity: (1)主函数 result%2 ==0 (2)主函数 if(result == 0) return 1 (3)sum digit 函数把 int temp = num %10 和 num = num/10 换一下顺序

countProduct: 改成 j< size

countElement: 改成 i< len; i++, 后面 arr[i+1] 改成 arr[i]

ArmstrongNumber: Math.pow(remainder, digitcount)

*reverseNum: reverseNum = reverseNum * 10 +i remainder*

KtimeElement : while loop 结束后 check if(count == k) res = element

distinctNumber* == 改成 !=

sumDistinct* 把排序移到最前

reverseString 的题来着 不是 str+=

eliminateVowel* default case 不需要 i++

checkPalindrome* 改成 result==tem

median* 把 i 改成 i-size

countDays* 建议上网看看闰年的定义

elementRange* 把&&改成|

getDigitSumParity: 1. 返回的判断条件写的都不对；2. getSum 里面 temp = 的顺序要提前

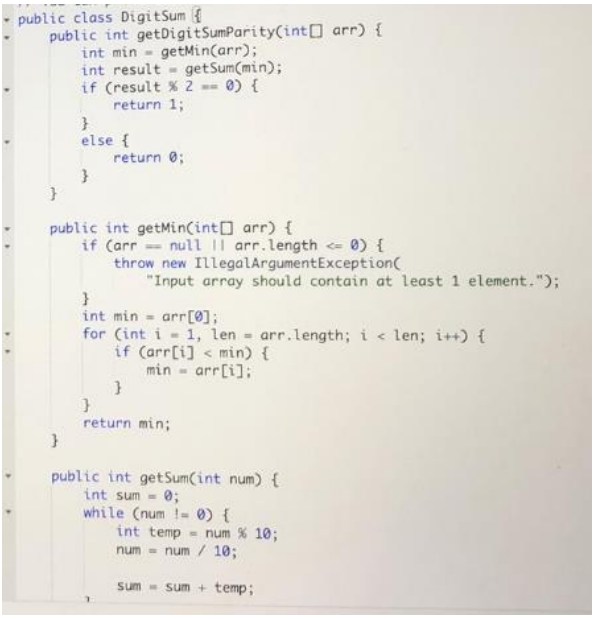
1.1 判断是否为偶数的条件 result%2!=0 应该改为 result%2==0

1.2 当 result 为 0 时，它返回了 0，应该返回 1 才对

(1)主函数 if (result % 2 ==0)

(2)主函数 if (result == 0) return 1

(3)sum digit函数把 int temp = num %10 和 num = num/10 换一下顺序



```
public class DigitSum {
    public int getDigitSumParity(int[] arr) {
        int min = getMin(arr);
        int result = getSum(min);
        if (result % 2 == 0) {
            return 1;
        }
        else {
            return 0;
        }
    }

    public int getMin(int[] arr) {
        if (arr == null || arr.length <= 0) {
            throw new IllegalArgumentException(
                "Input array should contain at least 1 element.");
        }
        int min = arr[0];
        for (int i = 1, len = arr.length; i < len; i++) {
            if (arr[i] < min) {
                min = arr[i];
            }
        }
        return min;
    }

    public int getSum(int num) {
        int sum = 0;
        while (num != 0) {
            int temp = num % 10;
            num = num / 10;

            sum = sum + temp;
```

replaceValues：把判断条件改成==1 就行

Occurrence: 给 while loop 后加个 i++;

GradingSystem：把||改成&&,

sortArray 把 if 条件的>改成<,

printPattern： 给 for loop 都加括号,

selectionsort：if 条件那里把 arr[x]改成 arr[y]

1. countOccurrence: while 循环里加上 i++
2. printPattern: 两个 for loop 括号加上
3. checkGrade: ||改成&&
4. reverseArray: +=改成-=
5. descendingSortArray: >改成<
6. drawPrintPattern: 第二个 for loop 后面分号删掉
7. selectionSortArray: 条件里面的 array[x]改成 array[y]

`sumDistinct:` 把 `sort` 提前

```java
import java.util.Arrays;
class Distinct
{
    int sumDistinct (int size, int[] inputArray)
    {
        Arrays.sort(inputArray);
        int sum =inputArray[0];

        int point = inputArray[0];
        for(int i = 1 ; i < size ; i++)
        {
            if(point != inputArray[i])
            {
                sum+=inputArray[i];
                point = inputArray[i];
            }

        }
        return sum;
    }
}
```

vowelsString: while 循环 i++放外面，最后的返回判断 len % 2 改成 len / 2

```java
public class StringCheck {

    private static final Set<Character> VOWELS = new HashSet<>();
    static {
        VOWELS.add('a');
        VOWELS.add('e');
        VOWELS.add('i');
        VOWELS.add('o');
        VOWELS.add('u');
        VOWELS.add('A');
        VOWELS.add('E');
        VOWELS.add('I');
        VOWELS.add('O');
        VOWELS.add('U');
    }

    public int vowelsString(String inputstr) {
        if (inputstr == null) {
            return 0;
        }
        int i = 0, vcount = 0, len = inputstr.length();
        while (i < len) {
            if (VOWELS.contains(inputstr.charAt(i))) {
                vcount += 1;

            }
            i++;
        }
        if (vcount > (len / 2))
            return 1;
        else
```

checkPalindrome: result = result * 10 + remainder, 最后返回要比较 temp

```
class Solution
{
    boolean checkPalindrome (int num)
    {
        int result=0;
        int remainder = 0;
        int temp = num;
        while(num != 0)
        {
            remainder = num % 10;
            result = result * 10 + remainder;
            num=num/10;
        }
        return (result == temp);
    }
}
```

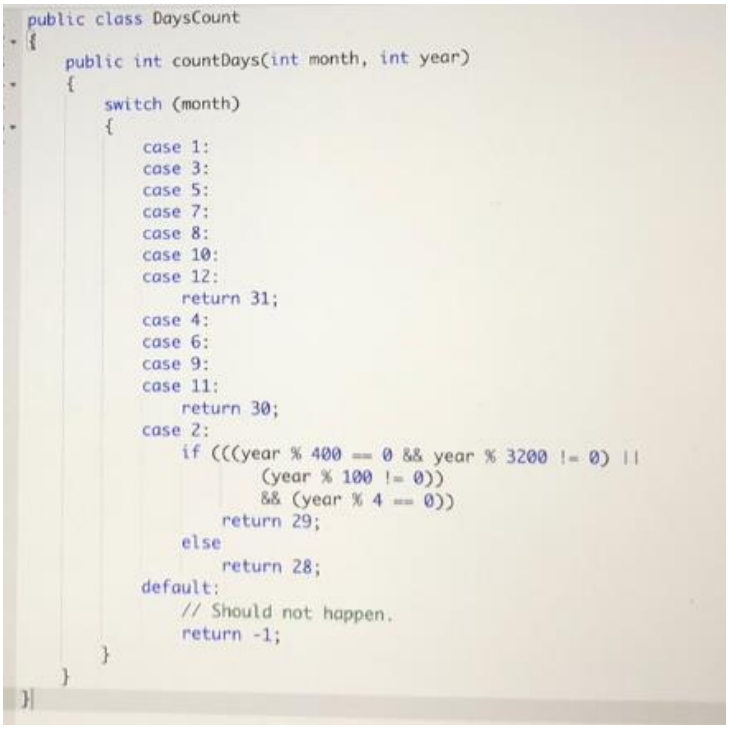
matrixSum: i++拿到内循环外面

```
public class Matrix {
    public int matrixSum(int[][] matrix) {
        int m = matrix.length;
        int n = matrix[0].length;
        int i = 0, j, sum = 0;
        while (i < m) {
            j = 0;
            while (j < n) {
                sum += matrix[i][j++];
            }
            i++;
        }
        return sum;
    }
}
```

atrixSum: while(j < n) sum += matrix[i][j++]; put i++ out of the while(j < n) loop

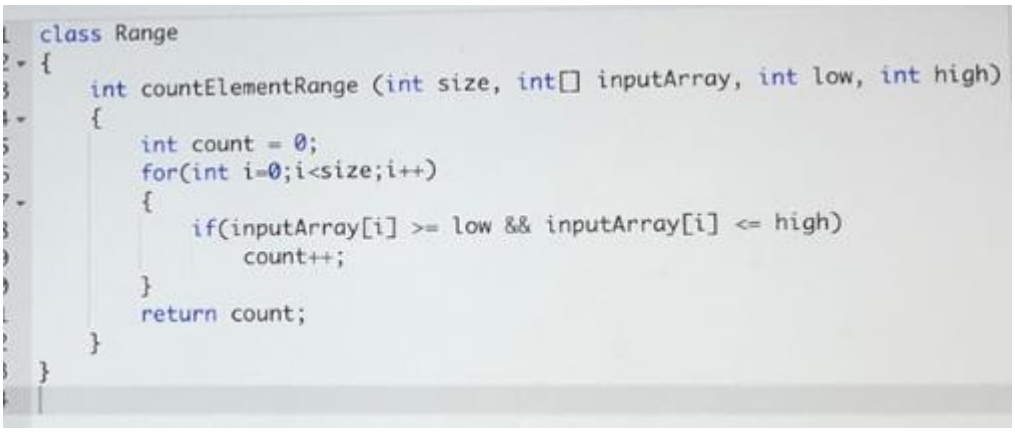
countDays: 参照 wiki 上的写法，按照 note 写只能过 11 个 case（总共 12 个）
，((y % 4) == 0 || (y % 100) != 0)应改为((y % 4) == 0 && (y % 100) != 0)

```
(year % 4 == 0) && (year %100 != 0) || (year % 400 == 0)
```



```
public class DaysCount
{
    public int countDays(int month, int year)
    {
        switch (month)
        {
            case 1:
            case 3:
            case 5:
            case 7:
            case 8:
            case 10:
            case 12:
                return 31;
            case 4:
            case 6:
            case 9:
            case 11:
                return 30;
            case 2:
                if (((year % 400 == 0 && year % 3200 != 0) ||
                        (year % 100 != 0))
                        && (year % 4 == 0))
                    return 29;
                else
                    return 28;
            default:
                // Should not happen.
                return -1;
        }
    }
}
```

countElementRange: || -> &&



```
class Range
{
    int countElementRange (int size, int[] inputArray, int low, int high)
    {
        int count = 0;
        for(int i=0;i<size;i++)
        {
            if(inputArray[i] >= low && inputArray[i] <= high)
                count++;
        }
        return count;
    }
}
```

ApearsKTimes

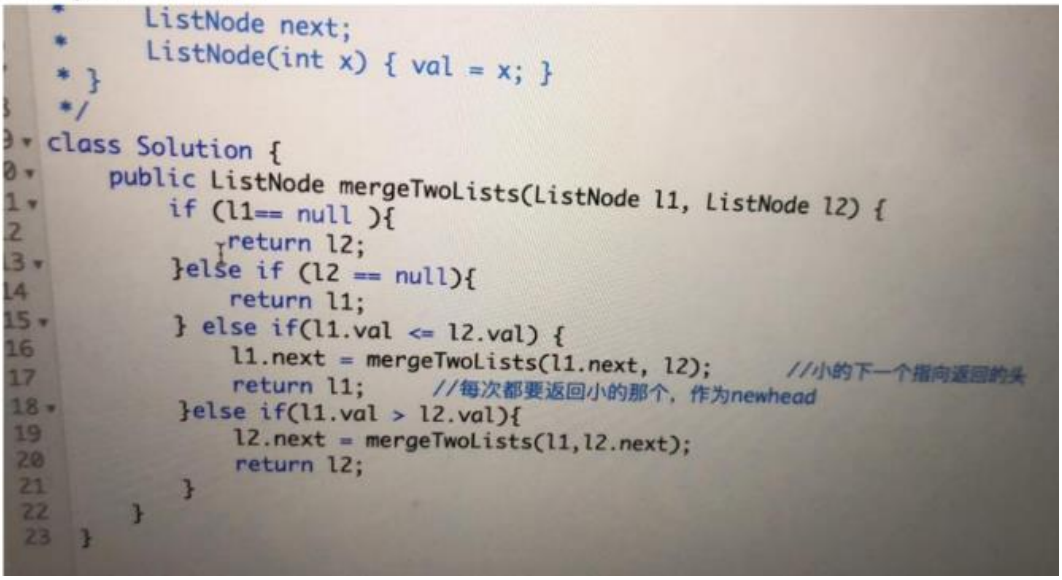
return 前 check 一下指针是不是在 array 的最后

```
if (input[size - 1] == element && count++ == k) {
            res = element;
        }
```

4. **AppearsK**: 1. add follow code after while loop: if (count == k) res = element;
   2. switch contents in if and else, judge if(element != inputArray[i]) first

Mergetwolists

16. **mergeTwoLists**:



```
 *      ListNode next;
 *      ListNode(int x) { val = x; }
 * }
 */
class Solution {
    public ListNode mergeTwoLists(ListNode l1, ListNode l2) {
        if (l1== null ){
            return l2;
        }else if (l2 == null){
            return l1;
        } else if(l1.val <= l2.val) {
            l1.next = mergeTwoLists(l1.next, l2);       //小的下一个指向返回的头
            return l1;      //每次都要返回小的那个，作为newhead
        }else if(l1.val > l2.val){
            l2.next = mergeTwoLists(l1,l2.next);
            return l2;
        }
    }
}
```

selection sort

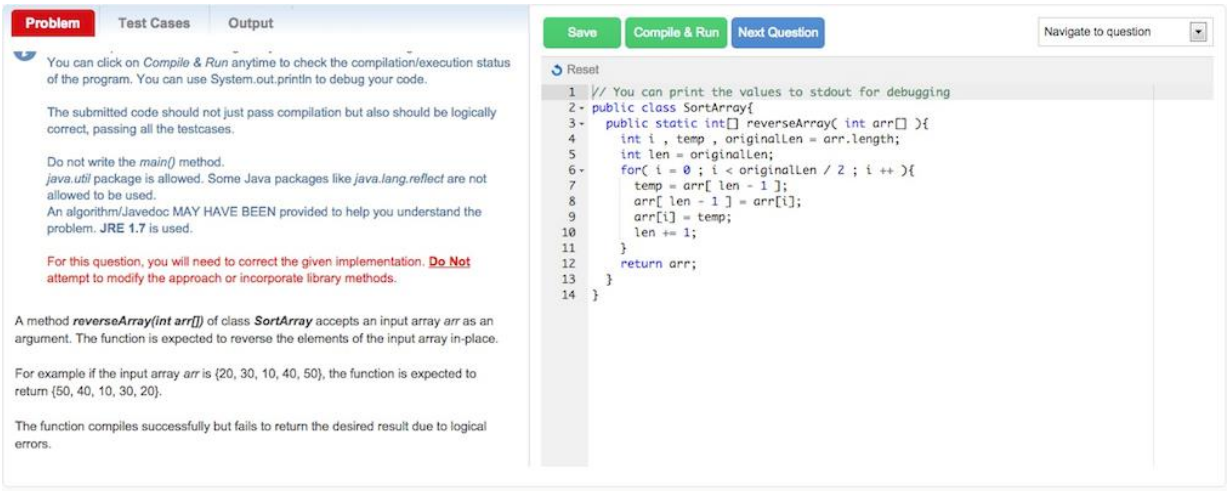
12. Selection sort

bug: if里面的判断它写的是arr[min]> arr[x],改成arr[min]> arr[y]

```
for (x = 0; x < n; x++) {
    int index_of_min = x;
    for (y = x; y < n; y++) {
        if (arr[index_of_min] > arr[x]) {
            y = index_of_min;
        }
        int temp = arr[x];
        arr[x] = arr[index_of_min];
        arr[index_of_min] = temp;
    }
}
// selection sort in ascending order 升序
for (int x = 0; x < n; x++) {
    int index_of_min = x;
    for (int y = x; y < n; y++) {
        if (arr[index_of_min] > arr[y]) {
            index_of_min = y;
         }
    }
    int temp = arr[x];
    arr[x] = arr[index_of_min];
    arr[index_of_min] = temp;
}
// bubble sort in descending order 降序
for (int x = 0; x < n; x++) {
    for (int y = x; y < n; y++) {
        if (arr[x] < arr[y]) {
            int temp = arr[x];
            arr[x] = arr[y];
            arr[y] = temp;
        }
    }
}
```

24. reverseArray

改法 1： += 改成 -=



```java
/**
 * Created by yaqunyu on 10/21/16.
 */
public class SortArray {
    public static int[] reverseArray(int arr[]){
        int i, temp, originallen = arr.length;
        int len = originallen;
        for(i =0; i<originallen/2; i++){
            System.out.println("i is: "+ i);
            //System.out.println("arr[i] is: "+ arr[i]);
            temp = arr[len-1];
            arr[len-1] = arr[i];
           // System.out.println("arr[len-1] is: "+ arr[len-1]);
            arr[i]=temp;
            //System.out.println("after reverse arr[i] is: "+ arr[i]);
            len -= 1;
            //System.out.println("len is: "+ len);
        }
        return arr;
    }
    public static void main(String[] args){
        int[] myIntArray = new int[]{20,2,3,5,9,6,16};
        System.out.println("Original Array is [20,2,3,5,9,6,16]");
        System.out.println("After revers array is: "+ Arrays.toString(reverseArray(myIntArray)));

    }
}
```

改法 2： if (arr[min] > arr[y]) for 循环中 arr[len - 1]改成 arr[len-i-1], 并且去掉 len+=1;

for(i=0; i<len/2; i++) { temp=arr[len-1-i]; arr[len-1-i]=arr[i]; arr[i]=temp; }

```java
import java.util.Arrays;

/**
 * Created by yaqunyu on 10/21/16.
 */
public class SortArray {
    public static int[] reverseArray(int arr[]){
        int i, temp, originallen = arr.length;
        int len = originallen;
        for(i =0; i<originallen/2; i++){
            System.out.println("i is: "+ i);
            //System.out.println("arr[i] is: "+ arr[i]);
            temp = arr[len-1-i];
            arr[len-1-i] = arr[i];
           // System.out.println("arr[len-1] is: "+ arr[len-1]);
            arr[i]=temp;
            //System.out.println("after reverse arr[i] is: "+ arr[i]);
            //len -= 1;
            //System.out.println("len is: "+ len);
        }
        return arr;
    }
    public static void main(String[] args){
        int[] myIntArray = new int[]{20,2,3,5,9,6,16};
        System.out.println("Original Array is [20,2,3,5,9,6,16]");
        System.out.println("After revers array is: "+ Arrays.toString(reverseArray(myIntArray)));

    }
}
```

25. drawPrintPattern

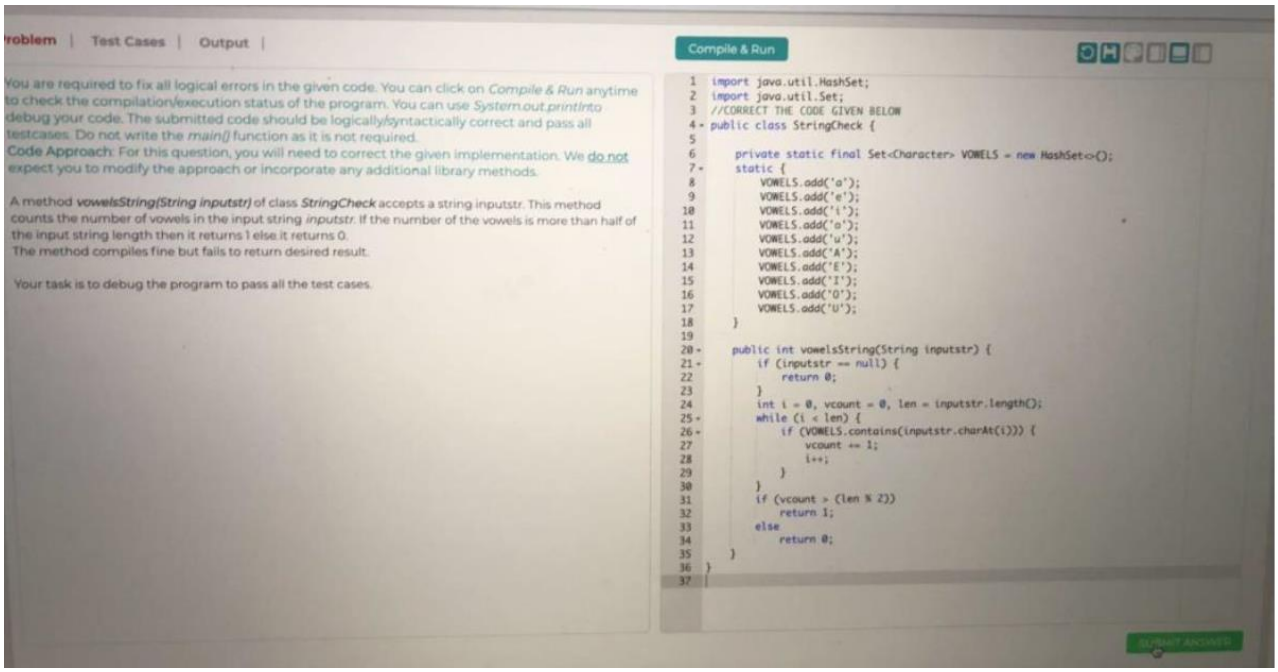
第二二个 for loop 后面面的分号删掉

26. vowelsString vowelString

while 循环 i++ 放到外面面

while 循环少了了 i++？？

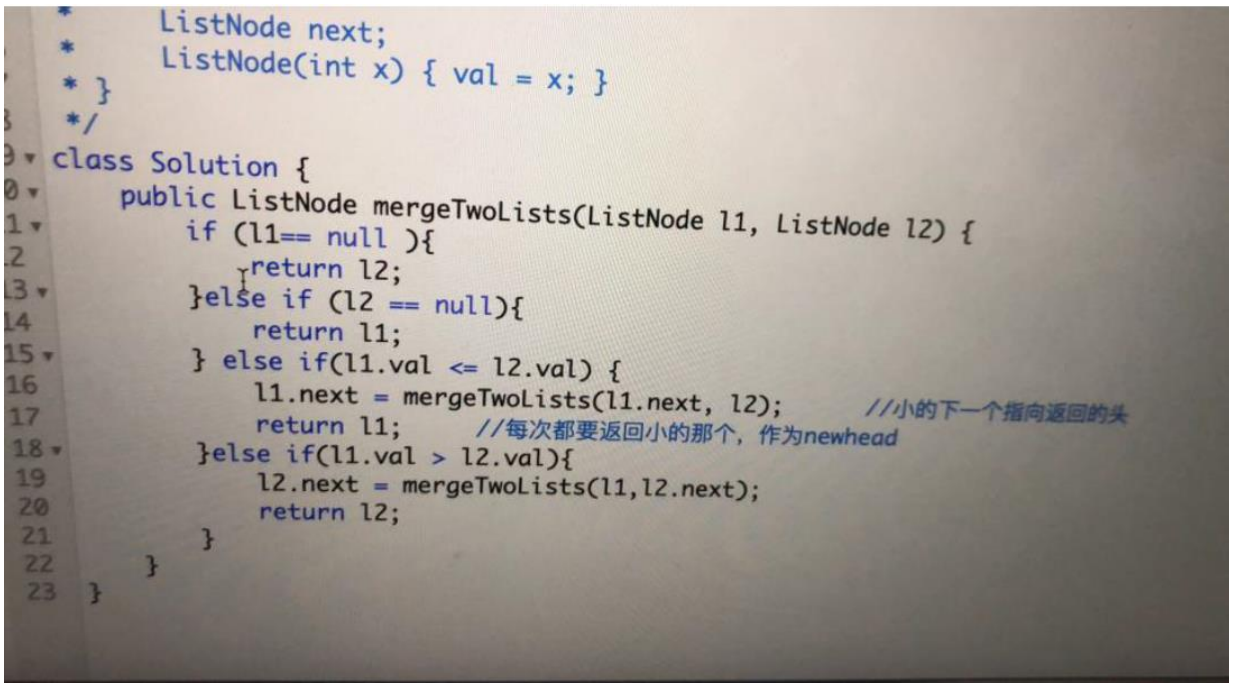
if (vcount > len % 2) 改成 if (vcount > (len / 2))



27. mergeTwoLists

```
else if (l1.val <= l2.val){
        l1.next = mergeTwoLists(l1.next, l2);
        return l1;

}
```



28. arraySum

计算一一个整数数组的和

sum = arr[i] 改成 sum += arr[i]

29. Remove element

for(i = index; i<len-1; i++){ arr[i]=arr[i++] bug!!! } 改为 arr[i] = arr[i+1]

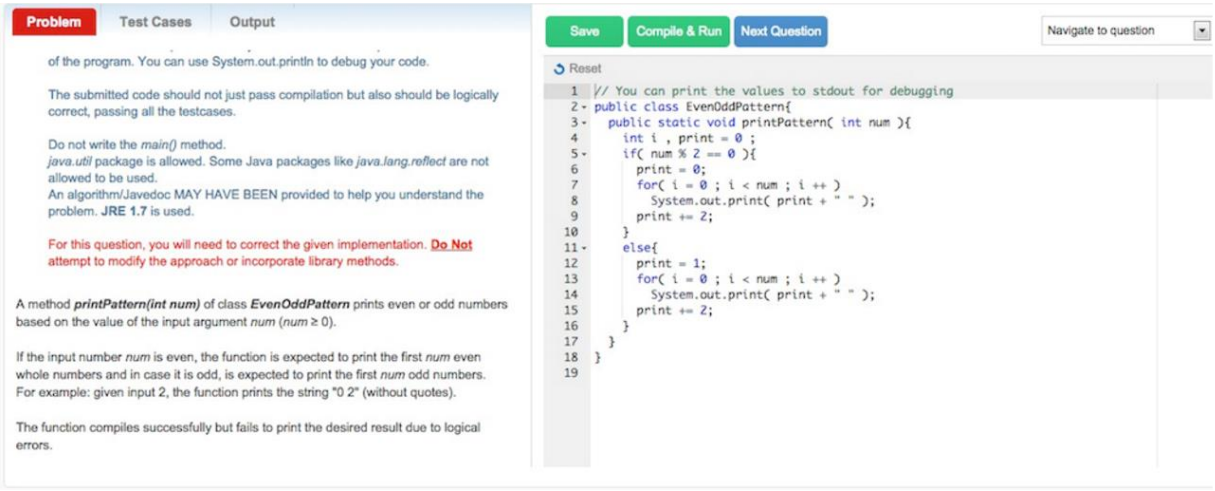

```java
import java.util.Arrays;

/**
 * Created by yaqunyu on 10/21/16.
 */
public class ShortArray {
    public static int[] removeElement(int arr[], int index) {
        int i, j, len = arr.length;
        if (index < len) {
            for (i = index; i < len - 1; i++) {
                System.out.println("Index now is: " + i);
                //if arr[i] = arr[i++] then after asign value, i will automatically increase 1
                // in our case, index = 3, after arr[3] = arr[4] i = 5
                arr[i] = arr[i+1];
                System.out.println("arr[i] is: " + arr[i]);
            }
            System.out.println("Array now is: " + Arrays.toString(arr));
            int rarr[] = new int[len - 1];
            for (i = 0; i < len - 1; i++)
                rarr[i] = arr[i];
            return rarr;
        } else
            return arr;
    }

    public static void main(String[] args) {
        int[] myIntArray = new int[]{20, 2, 3, 5, 9, 6, 16};
        System.out.println("Original Array is [20,2,3,5,9,6,16]");
        System.out.println("After revers array is: " + Arrays.toString(removeElement(myIntArray, 3)));
    }
}
```

30. Even odd pattern

even 和 odd 处的 for loop 缺少大大括号




```java
/**
 * Created by yaqunyu on 10/21/16.
 */
public class EvenOddPattern {
    public static void printPattern(int num) {
        int i, print =0;
        if(num % 2 == 0){
            print = 0;
            for(i = 0; i<num; i++) {
                System.out.print(print + " ");
                print += 2;
            }
        }
        else {
            print = 1;
            for (i = 0; i < num; i++) {
                System.out.print(print + " ");
                print += 2;
            }
        }
    }
    public static void main(String[] args){
        printPattern(2);
    }
}
```

31. Manchester

这个有两处错误，是要判断 array 里每位数字和前面一位是不是相等，index0 的数字单独和 0 判断一下，所以 0 那句不能直接 res[0] = 0 要 res[0] = (arr[0] != 0) ? 1:0，后面 for 循环里判断是要把 arr == arr[i-1]改成 arr != arr[i-1]

Manchester Array: 给一个 input array, 如果 A[i] != A[i-1], return 1，反之 return 0, 需要 output 另一个 array
result = (A[i] == A[i-1]); => result = A[i] != A[i-1];
output[i] = result ? 1 : 0;
ret[0]也要加一下，不然有一个 case 过不了

32. Remove duplicate from unsorted array

报错是 Index out of bound, 因为 for 循环里有 k<length, 下面却使用 arr[k+1] 改为 k < length - 1

33. Remove duplicates

循环下标从 i+1 开始

34. Insertion sort

if (arr[i-1] > arr[i])

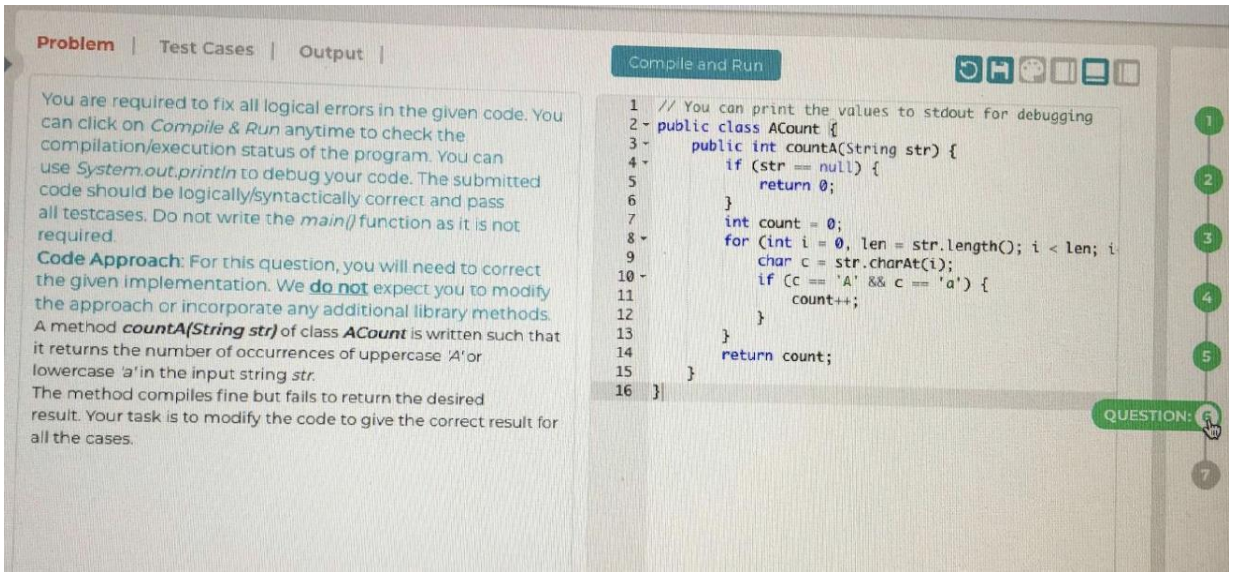
while (j > 0 && arr[j - 1] > temp)

insertion sort < 改 > 或者 > 改 <

```
for (int i = 1; i < n; i++) {
    if (arr[i - 1] > arr[i]) {
        int temp = arr[i];
        int j = i;
        while (j > 0 && arr[j - 1] > temp) {
            arr[j] = arr[j - 1];
            j--;
        }
        arr[j] = temp;
    }
}
```

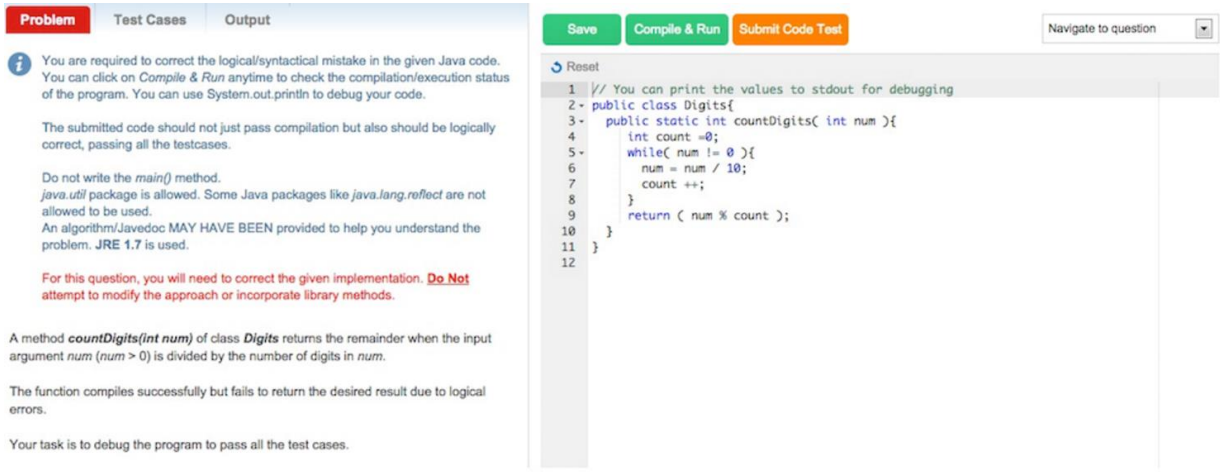
35. countA

if (c == 'A' || c == 'a')



36. countDigits

这里面到最后 num 直接是 0，造成 bug，需要用一个数来储存 num 和执行运算



```java
*/
//function is to dived num by its count
public class Digits {
    public static int countDigits(int num){
        int count = 0;
        int temp = num;
        while(temp != 0){
            //to check the how many digit is num
            //however if num = num/10 then at the last step of while, the num will always be 0
            //so that is the error
            temp = temp/10;
            count++;
        }
        //use operator % will get the result of remind
        return (num%count);
    }
    public static void main(String[] args){
        System.out.println("Remain result for count digit 235 is: "+countDigits(235));
    }
}
//System.out.println("count is: "+ count);
//System.out.println("temp is: "+ temp + " num is: "+num);
//System.out.println("result is: "+ num/count);
```

37.arrayOperation

<和>写反了