Amazon oa1

Tuesday, October 29, 2019 8:35 PM

1. Arraysort, descending order:
   a. 题1 ：if (max > arr[j]) 改成 <
   b. 题2: if (arr[i] > arr[j]) 改成 <
2. Arraysort2, ascending:判断条件里的>换成<, 两个变量 small和pos 没有用到
3. countDigits
   a.
```
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
        //so that is the error
        temp = temp/10;
        count++;
    }
    //use operator % will get the result of remind
    return (num%count);
}
```
4. replaceValues: i<=len, j<=len改成 i<len, j<len
5. reverseArray: line10: len+=1改为len -= 1;
6. Remove element: line7: arr[i]=arr[i++] 改为 arr[i] = arr[i+1]
7. printPattern, evenOddPattern: 两个for缺少大括号,
   a.
```
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;
        }
    }
}
```

8. Manchester code : ????
   a. result=(a[i]==a[i-1])改成result=(a[i]!=a[i-1])
   b. ret[0] 也要加一下
   c. 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]*
9. Array: = arr[i] 改为 sum += arr[i]
10. printPattern,aababcabcd:
    a.
```
改为
char ch = 'a';
char print = ch;
for(int j = 0; j <=i; j++) {
    System.out.print((print++));
}
System.out.println("");
```
11. printPattern,1111:
    a.
```
public static void print3(int row) {
        int x = 1;
        for (int i = 1; i <= row; i++) {
            for (int j = i; j > 0; j--) {
                System.out.print(x + "" + x);
            }
            System.out.println();
        }
    }
```
    bug:第一层for循环少了大括号，导致最后那个System.out.println()最后只执行一次
12. removeDuplicates: 循环里改成 k<length-1
13. removeDupilicates from unsorted array:
    a. 循环下标从 i+1 开始

数

14. Array 奇数偶数，for循环里，i+=2 改为i++
15. insertionsort：
    a. 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;
    }
}
```

16. countOccurrence: 返回value在array中出现的次数，while里面最后加上i++
17. checkGrade: (x>=70)||(x<80)里面||改为&&
18. arrayOperation <> 反了。？？？？
19. selectionSort, ascending order, arr[index_of_min]>arr[x]改为arr[y]
20. distinctSum: == 改为 !=
21. sumDistinct: 初始化sum数组写在sort之后
22. eliminateVowel: default 去掉i++
23. checkPalindrome:
    a. 把最后返回值改成return(result==temp);
24. distinctElementCount: if(flag==1)改为if(flag==0)
    a. 判断是否重复的flag 把0/1写反了
25. matrixSum:
    a.

```
while(j < n)
        sum += matrix[i++][j++]
```

改成

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

26. labelProduct

a.

```
class Product
{
    public String labelProduct(String productID)
    {
        String str = "";
        int i = productID.length() - 1;
        productID = productID.toLowerCase();
        while(i >= 0)
        {
            if((productID.charAt(i) >= 'a') && (productID.charAt(i) <= 'z'
            {
                str+= productID.charAt(i);
            }
            i--;
        }

        return str;
    }
}
```

27. medianValue:

a. arr2 越界，改为 arr2[i-size]

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

28. countDays

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

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

29. countElementRange: || 改为&&

30. 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

31. countProduct

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

32. countElement:

1. 改成 i < len; i++

2. 后面面 arr[i+1] 改成 arr[i]

33. checkArmstrong:
    1. result+=math.pow(result, digitCount)应该改为 result+=math.pow(remainder, digitCount)
34. reverseNumber
    1. 应该是 reverseNum = reverseNum*10 + Reminder
35. appearsKtimes:
    1. 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])
36. reverseString:
    1. 去掉len++ ?
    2. 不是str+= ?
37. vowelsString:
    1. While 循环i++放到外面
    2. if (vcount > len % 2) 改成 if (vcount > (len / 2))
38. mergeTwoLists
    1.
```
else if (l1.val <= l2.val){
        l1.next = mergeTwoLists(l1.next, l2);
        return l1;

}
```
39. countA
    1. if (c == 'A' || c == 'a')
40.

```java
import java.util.Arrays;
class AppearsK
{
    public int appearsKTimes(int size, int inputArray[], int k)
    {
        Arrays.sort(inputArray);
        int i=1, count = 1;
        int element = inputArray[0];
        int res = -1;
        while(i < size)
        {
            if(element == inputArray[i])
            {
                count++;
            }
            else
            {
                if(count == k)
                {
                    res = element;
                }
                element = inputArray[i];
                count = 1;
            }
            i++;
        }
        if (count ==k)
        res = element;
        return res;
    }
}
```