```
                chas[end--] = 0;
            }
        }
```

字符串的统计字符串

【题目】

给定一个字符串 str，返回 str 的统计字符串。例如，"aaabbadddffc"的统计字符串为"a_3_b_2_a_1_d_3_f_2_c_1"。

【补充题目】

给定一个字符串的统计字符串 cstr，再给定一个整数 index，返回 cstr 所代表的原始字符串上的第 index 个字符。例如，"a_1_b_100"所代表的原始字符串上第 0 个字符是'a'，第 50 个字符是'b'。

【难度】

士 ★☆☆☆

【解答】

原问题。解决原问题的方法有很多，本书仅提供一种供读者参考。具体过程如下：

1．如果 str 为空，那么统计字符串不存在。

2．如果 str 不为空。首先生成 String 类型的变量 res，表示统计字符串，还有整型变量 num，代表当前字符的数量。初始时字符串 res 只包含 str 的第 0 个字符(str[0])，同时 num=1。

3．从 str[1]位置开始，从左到右遍历 str，假设遍历到 i 位置。如果 str[i]==str[i-1]，说明当前连续出现的字符(str[i-1])还没结束，令 num++，然后继续遍历下一个字符。如果 str[i]!=str[i-1]，说明当前连续出现的字符(str[i-1])已经结束，令 res=res+"_"+num+"_"+str[i]，然后令 num=1，继续遍历下一个字符。以题目给出的例子进行说明，在开始遍历"aaabbadddffc"之前，res="a"，num=1。遍历 str[1~2]时，字符'a'一直处在连续的状态，所以 num 增加到 3。遍历 str[3]时，字符'a'连续状态停止，令 res=res+"_"+"3"+"_"+"b"(即"a_3_b")，num=1。遍历 str[4]，字符'b'在连续状态，num 增加到 2。遍历 str[5]时，字符'a'连续状态停止，令 res 为"a_3_b_2_a"，num=1。依此类推，当遍历到最后一个字符时，res 为

"a_3_b_2_a_1_d_3_f_2_c"，num=1。

4．对于步骤 3 中的每一个字符，无论连续还是不连续，都是在发现一个新字符的时候再将这个字符连续出现的次数放在 res 的最后。所以当遍历结束时，最后字符的次数还没有放入 res，所以最后令 res=res+"_"+num。在步骤 3 的例子中，当遍历结束时，res 为 "a_3_b_2_a_1_d_3_f_2_c"，num=1，最后需要把 num 加在 res 后面，令 res 变为 "a_3_b_2_a_1_d_3_f_2_c_1"，然后再返回。

具体过程请参看如下代码中的 getCountString 方法。

```
public String getCountString(String str) {
        if (str == null || str.equals("")) {
                return "";
        }
        char[] chs = str.toCharArray();
        String res = String.valueOf(chs[0]);
        int num = 1;
        for (int i = 1; i < chs.length; i++) {
           if (chs[i] != chs[i - 1]) {
                res = concat(res, String.valueOf(num), String.valueOf(chs[i]));
                num = 1;
           } else {
                num++;
           }
        }
        return concat(res, String.valueOf(num), "");
}

public String concat(String s1, String s2, String s3) {
        return s1 + "_" + s2 + (s3.equals("") ? s3 : "_" + s3);
}
```

补充问题。求解的具体过程如下：

1．布尔型变量 stage，stage 为 true 表示目前处在遇到字符的阶段，stage 为 false 表示目前处在遇到连续字符统计的阶段。字符型变量 cur，表示在上一个遇到字符阶段时，遇到的是 cur 字符。整型变量 num，表示在上一个遇到连续字符统计的阶段时，字符出现的数量。整型变量 sum，表示目前遍历到 cstr 的位置相当于原字符串的什么位置。初始时，stage=true，cur=0（字符编码为 0 表示空字符），num=0，sum=0。

2．从左到右遍历 cstr，举例说明这个过程，cstr="a_100_b_2_c_4"，index=105。遍历完 str[0]=='a'后，记录下遇到字符'a'，即 cur='a'。遇到 str[1]=='_'，表示该转阶段了，从遇到字符的阶段变为遇到连续字符统计的阶段，即 stage=!stage。遇到 str[2]=='1'时，num=1；遇到 str[3]=='0'时，num=10；遇到 str[4]=='0'时，num=100；遇到 str[5]=='_'，表示遇到连续字

符统计的阶段变为遇到字符的阶段；遇到 str[6]=='b'，一个新的字符出现了，此时令 sum+=num（即 sum=100），sum 表示目前原字符串走到什么位置了，此时发现 sum 并未到达 index 位置，说明还要继续遍历，记录下遇到了字符'b'，即 cur='b'，然后令 num=0，因为字符'a'的统计已经完成，现在 num 开始表示字符'b'的连续数量。也就是说，每遇到一个新的字符，都把上一个已经完成的统计数 num 加到 sum 上，再看 sum 是否到达 index，如果已到达，就返回上一个字符 cur，如果没到达，就继续遍历。

3．每个字符的统计都在遇到新字符时加到 sum 上，所以当遍历完成时，最后一个字符的统计数并不会加到 sum 上，最后要单独加。

具体过程请参看如下代码中的 getCharAt 方法。

```
public char getCharAt(String cstr, int index) {
        if (cstr == null || cstr.equals("")) {
                return 0;
        }
        char[] chs = cstr.toCharArray();
        boolean stage = true;
        char cur = 0;
        int num = 0;
        int sum = 0;
        for (int i = 0; i != chs.length; i++) {
                if (chs[i] == '_') {
                        stage = !stage;
                } else if (stage) {
                        sum += num;
                        if (sum > index) {
                                return cur;
                        }
                        num = 0;
                        cur = chs[i];
                } else {
                        num = num * 10 + chs[i] - '0';
                }
        }
        return sum + num > index ? cur : 0;
}
```

判断字符数组中是否所有的字符都只出现过一次

【题目】

给定一个字符类型数组 chas[]，判断 chas 中是否所有的字符都只出现过一次，请根据以下不同的两种要求实现两个函数。