```
        int res = 0;
        int cur = 0;
        for (int i = posi ? 0 : 1; i < chas.length; i++) {
                cur = '0' - chas[i];
                if ((res < minq) || (res == minq && cur < minr)) {
                        return 0; // 不能转
                }
                res = res * 10 + cur;
        }
        if (posi && res == Integer.MIN_VALUE) {
                return 0; // 不能转
        }
        return posi ? -res : res;
}
```

替换字符串中连续出现的指定字符串

【题目】

给定三个字符串 str、from 和 to，把 str 中所有 from 的子串全部替换成 to 字符串，对连续出现 from 的部分要求只替换成一个 to 字符串，返回最终的结果字符串。

【举例】

str="123abc"，from="abc"，to="4567"，返回"1234567"。

str="123"，from="abc"，to="456"，返回"123"。

str="123abcabc"，from="abc"，to="X"，返回"123X"。

【难度】

士 ★☆☆☆

【解答】

解决本题的方法有很多。本书仅提供一种供读者参考。如果把 str 看作字符类型的数组，首先把 str 中 from 部分所有位置的字符编码设为 0（即空字符），比如，str="12abcabca4"，from="abc"，处理后 str 为['1','2',0,0,0,0,0,0,'a','4']。具体过程如下：

1. 生成整型变量 match，表示目前匹配到 from 字符串的什么位置，初始时，match=0。

2. 从左到右遍历 str 中的每个字符，假设当前遍历到 str[i]。

3．如果 str[i]==from[match]。如果 match 是 from 最后一个字符的位置，说明在 str 中发现了 from 字符串，则从 i 位置向左的 M 个位置，都把字符编码设为 0，M 为 from 的长度，设置完成后令 match=0。如果 match 不是 from 最后一个字符的位置，令 match++。继续遍历 str 的下一个字符。

4．如果 str[i]!=from[match]，说明匹配失败，令 match=0，即回到 from 开头重新匹配。继续遍历 str 的下一个字符。

通过上面的过程，接下来替换就比较容易，比如['1','2',0,0,0,0,0,0,'a','4']，将不为 0 的区域拼在一起，连续为 0 的部分用 to 来替换，即"12"+to+"a4"即可。

全部过程请参看如下代码中的 replace 方法。

```
public String replace(String str, String from, String to) {
        if (str == null || from == null || str.equals("") || from.equals("")) {
                return str;
        }
        char[] chas = str.toCharArray();
        char[] chaf = from.toCharArray();
        int match = 0;
        for (int i = 0; i < chas.length; i++) {
                if (chas[i] == chaf[match++]) {
                        if (match == chaf.length) {
                                clear(chas, i, chaf.length);
                                match = 0;
                        }
                } else {
                        match = 0;
                }
        }
        String res = "";
        String cur = "";
        for (int i = 0; i < chas.length; i++) {
                if (chas[i] != 0) {
                        cur = cur + String.valueOf(chas[i]);
                }
                if (chas[i] == 0 && (i == 0 || chas[i - 1] != 0)) {
                        res = res + cur + to;
                        cur = "";
                }
        }
        if (!cur.equals("")) {
                res = res + cur;
        }
        return res;
}

public void clear(char[] chas, int end, int len) {
        while (len-- != 0) {
```

```
                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