题目描述：
【敏感字段加密】给定一个由多个命令字组成的命令字符串：
1、字符串长度小于等于 127 字节，只包含大小写字母，数字，下划线和偶数个双引号；
2、命令字之间以一个或多个下划线_进行分割；
3、可以通过两个双引号""来标识包含下划线_的命令字或空命令字（仅包含两个双引号的命令字），双引号不会在命令字内部出现；
请对指定索引的敏感字段进行加密，替换为******（6 个*），并删除命令字前后多余的下划线_。如果无法找到指定索引的命令字，输出字符串 ERROR。

输入描述：
输入为两行，第一行为命令字索引 K（从 0 开始），第二行为命令字符串 S。
输出描述：
输出处理后的命令字符串，如果无法找到指定索引的命令字，输出字符串 ERROR
补充说明：

```
示例1
输入：1
     password__a12345678_timeout_100
输出：password_******_timeout_100
说明：

示例2
输入：2
     aaa_password_"a12_45678"_timeout__100_""_
输出：aaa_password_******_timeout_100_""
说明：
```

```java
import java.util.Scanner;

// 注意类名必须为 Main, 不要有任何 package xxx 信息
public class Main {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        // 注意 hasNext 和 hasNextLine 的区别
        while (in.hasNextLine()) { // 注意 while 处理多个 case
            int k = Integer.parseInt(in.nextLine());
            String s = in.nextLine();
            System.out.println(generate(s, k));
        }
    }

    public static String generate(String s, int k) {
        StringBuilder res = new StringBuilder();
        int l = 0, r = 1, cnt = 0;
        while (l < s.length() && s.charAt(l) == '_') l++;
        r = l;
        if (l == s.length()) return "ERROR";
        while (r < s.length()) {
            if (cnt == k) {
                if (r < s.length() && s.charAt(r) == '"') {
                    r++;
                    while (r < s.length() && s.charAt(r) != '"') {
                        r++;
                    }
                    r++;
                    res.append("*******");
                } else {
                    while (r < s.length() && s.charAt(r) != '_') r++;
                    res.append("*******");
                }
            } else {
                if (r < s.length() && s.charAt(r) == '"') {
                    r++;
                    while (r < s.length() && s.charAt(r) != '"') {
                        r++;
                    }
                    r++;
                } else {
                    while (r < s.length() && s.charAt(r) != '_') r++;
                }
                res.append(s.substring(l, r));
            }
            while (r < s.length() && s.charAt(r) == '_') r++;
            if (r < s.length()) res.append("_");
            l = r;
            cnt++;
        }
        return cnt > k ? res.toString() : "ERROR";
    }
}
```