

题目描述：

给定用户密码输入流 *input*，输入流中字符‘<’表示退格，可以清除前一个输入的字符，请你编写程序，输出最终得到的密码字符，并判断密码是否满足如下的密码安全要求。

密码安全要求如下：

1.密码长度 ≥ 8 ;

2.密码至少需要包含 1 个大写字母;

3.密码至少需要包含 1 个小写字母;

4.密码至少需要包含 1 个数字;

5.密码至少需要包含 1 个字母和数字以外的非空白特殊字符

注意空串退格后仍然为空串，且用户输入的字符串不包含‘<’字符和空白字符。

输入描述：

用一行字符串表示输入的用户数据，输入的字符串中‘<’字符标识退格，用户输入的字符串不包含空白字符，例如：

ABC<c89%000<

输出描述：

输出经过程序处理后,输出的实际密码字符串,并输出改密码字符串是否满足密码安全要求。

两者间由‘,’分隔， 例如：

```
ABc8 9%00,true
```

示例 1

输入：

```
ABC
```

输出：

```
ABc89%00,true
```

说明：

解释：多余的 *C* 和 *O* 由于退格被去除,最终用户输入的密码为 *ABc8 9%00*，且满足密码安全要求，输出 *true*

示例 2

输入：

输出：

```
ABC,false
```

说明：

不满足密码安全要求

示例 3

输入：

```
AB<
```

输出：

```
,false
```

说明：

输出字符串为空，不满足密码安全要求

```
package main
```

```

import (
    "fmt"
    "strings"
)

func main() {
    var s string
    fmt.Scanln(&s)
    length := len(s)
    res := []string{}
    lch,bch,num,tzf:= false,false,false,false
    for i:=0; i<length; i++){
        if s[i] >= 'a' && s[i] <='z'{
            lch = true
        }else if s[i] >= 'A' && s[i] <='Z'{
            bch = true
        }else if s[i] >= '0' && s[i] <='9'{
            num = true
        }else if s[i] == '<'{
            if len(res)>0 {
                res = res[:len(res)-1]
            }
            continue
        }else {
            tzf = true
        }
        res = append(res, string(s[i]))
    }
    if len(res) >= 8 && lch && bch && num && tzf {
        fmt.Printf("%s,%v",strings.Join(res,""),true)
    }else {
        fmt.Printf("%s,%v",strings.Join(res,""),false)
    }
}

```