

一、编程题

ACM：密码输入检测

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：

ABc89%00,true

补充说明：

示例1

输入：ABC

输出：ABc89%00,true

说明：解释：多余的C和0由于退格被去除,最终用户输入的密码为ABc89%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

代码：

```
#include <bits/stdc++.h>
```

```
#include <cctype>
```

```
#include <ratio>
```

```
using namespace std;
```

```
int main() {
```

```
    string str;
```

```
    cin >> str;
```

```
    vector<char> vec;
```

```
    vector<bool> flag(4, false);
```

```
    bool have_upper = false;
```

```
    bool have_lower = false;
```

```
    bool have_digit = false;
```

```
    bool have_other = false;
```

```
    for(const auto& item : str) {
```

```

        if(!vec.empty() && item == '<') {
            vec.pop_back();
        }else if(item != '<'){
            vec.push_back(item);
            if(islower(item)) {
                have_lower = true;
            }else if(isupper(item)) {
                have_upper = true;
            }else if(isdigit(item)){
                have_digit = true;
            }else {
                have_other = true;
            }
        }
    }
    string s(vec.begin(), vec.end());
    string status = (have_lower && have_upper && have_digit && have_other && s.size() >= 8) ?
    "true" : "false";
    cout << s << "," << status;
    return 0;
}
// 64 位输出请用 printf("%lld")

```