

题目描述：

警察在侦破一个案件时，得到了线人给出的可能犯罪时间，形如 “**HH:MM**” 表示的时刻。

根据警察和线人的约定，为了隐蔽，该时间是修改过的，解密规则为：利用当前出现过的数字，构造下一个距离当前时间最近的时刻，则该时间为可能的犯罪时间。每个出现数字都可以被无限次使用。

输入描述：

形如 **HH:SS** 的字符串，表示原始输入

输出描述：

形如 **HH:SS** 的字符串，表示推理出来的犯罪时间

补充说明：

1. 可以保证线人给定的字符串一定是合法的。例如，“**01:35**” 和 “**11:08**” 是合法的，“**1:35**” 和 “**11:8**” 是不合法的。
2. 最近的时刻有可能在第二天。

示例 1

输入：

18:52

输出：

18:55

说明：

利用数字 **1, 8, 5, 2** 构造出来的最近时刻是 **18:55**，是 **3** 分钟之后。结果不是 **18:51** 因为这个时刻是 **23** 小时 **59** 分钟之后。

示例 2

输入：

23:59

输出：

22:22

说明：

利用数字 **2, 3, 5, 9** 构造出来的最近时刻是 **22:22**。 答案一定是第二天的某一时刻，所以选择可构造的最小时刻为犯罪时间。

```
import java.util.*;
```

```
// 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 input = new Scanner(System.in);
```

```
        String str = input.next();
```

```
        input.close();
```

```
        List<Integer> num = new ArrayList();
```

```
        for (int i = 0; i < str.length(); i++) {
```

```
            if (str.charAt(i) != ':') {
```

```
                num.add(str.charAt(i) - '0');
```

```
            }
```

```
        }
```

```
        int totalMin = (num.get(0) * 10 + num.get(1)) * 60 + (num.get(2) * 10 + num.get(3));
```

```
        int minInterval = 24 * 60 * 60;
```

```
        String result = getResult(num, totalMin, minInterval);
```

```
        System.out.println(result);
```

```
    }
```

```
    public static String getResult(List<Integer> num, int totalMin, int minInterval) {
```

```
        String result = "";
```

```
        for (int i0 = 0; i0 < num.size(); i0++) {
```

```
            if (num.get(i0) > 2)
```

```
                continue;
```

```
            for (int i1 = 0; i1 < num.size(); i1++) {
```

```
                if (num.get(i0) == 2 && num.get(i1) > 3)
```

```
                    continue;
```

```
                for (int i2 = 0; i2 < num.size(); i2++) {
```

```
                    if (num.get(i2) > 5)
```

```
                        continue;
```

```
                    for (int i3 = 0; i3 < num.size(); i3++) {
```

```
                        int newTime = (num.get(i0) * 10 + num.get(i1)) * 60 + (num.get(i2) * 10 + num.get(i3));
```

```

                                i2) * 10 + num.get(i3));
if (newTime < totalMin) {
    newTime += 24 * 60 * 60;
} else if (newTime == totalMin) {
    continue;
}
if (newTime - totalMin < minInterval) {
    minInterval = newTime - totalMin;
    result = num.get(i0) + String.valueOf(num.get(i1)) + ":" +
num.get(
                                i2) + num.get(i3);
}
}
}
}
}
return result;
}
}

```