

# 权限的修改

## 限制

- 1000 ms
- 32767 KB

在 Linux 操作系统中文件的权限可以通过 `rwX` 的方式来表示，`r` 表示可读，`w` 表示可写，`x` 表示可执行。

在操作系统实现这个功能时，采用的方式是按位存储的。当我们不给任何权限时，存储表现为三个二进制位 `000`，如果可读则第一位会变成 `1`、如果可写则第二位会变成 `1`、如果可执行则第三位会变成 `1`。例如

- 如果可读、可写、可运行，就表示为二进制的 `111`，转换成十进制就是 `7`。
- 如果可读、可写、不可运行，就表示为二进制的 `110`，转换成十进制就是 `6`。
- 如果可读、不可写、可运行，就表示为二进制的 `101`，转换成十进制就是 `5`。

在这题中，我们会得到文件当前的权限和希望添加或删除的权限（如果已经对存在/不存在的权限添加/删除则不做改变）。你需要输出最终改变后的权限的十进制描述。

提示 1: `r`，`w`，`x` 的顺序是不会改变的，不会出现 `r` 出现在 `w` 后或 `x` 出现在 `w` 之前一类的情况。

提示 2: 对于不知道读入有多少行的情况，可以用：

```
while (scanf(/* 这部分省略 */) != EOF) {  
}
```

的方式进行。

## 输入格式

输入第一行是当前的文件权限，用 `r`、`w`、`x` 的字母中的一个或多个来表示（`r`，`w`，`x` 不会重复）。之后每一行会以 `+` 或 `-` 开头；如果以 `+` 开头，则表示添加一个权限，如果 `-` 开头，则表示删除一个权限。

## 输出格式

将修改后的权限以十进制数字的方式输出出来。

## 习题目的

- 练习对字符与字符串的操作和使用
  - 鼓励位运算
  - 练习使用 `EOF` 表示读到文件末尾
- 

## 样例输入 1

```
rw  
+x
```

## 样例输出 1

```
7
```

---

## 样例输入 2

```
w  
+x  
+r  
-w
```

## 样例输出 2

```
5
```

---

## 提示信息

请注意，包括最后一行在内的输入数据每一行，都以换行符结尾，

你的程序会被测评机输入各种不同的，但是符合描述的输入数据（所以你需要用 `scanf` 接受输入）。它之后会对你的输出和预期的标准输出进行比对，如果你在任何一个给出的输入上都给出了正确的输出，你就会通过这一测试。

同时，请注意，在测试中使用样例输入得到正确的样例输出结果也不一定代表程序就完备了。提交前，你最好再自己多设计几组符合题目要求的输入然后计算得出结果，与输出值比较，看看是不是每一次都相同。

请注意，不要额外输出任何的内容。任何所谓的“友好提示信息”，在测评机来看都是由于程序设计错误而输出的错误内容。