

日期合法性

限制

- 1000 ms
 - 32767 KB
-

你已经和老师一起看清了很多程序语言中的“是非”，这下，你又需要在没有老师的任何帮助下完成一个任务啦。

这次的题目也不是很难。测试程序将随机给你一个格式为 `YYYY-MM-DD` 连接了年、月、日的日期字符串（其中的数字都是没有前导零的，也就是说 7 不会被写成 07）。你将需要判断这个日期在我们的公历体系内是否真实存在。如果存在则输出 `YES`，否则输出 `NO`（注意输出的字母都一定是大写字母）。

测试程序给出的测试输入数据中，年、月、日都是正整数（也就是大于零的整数），均不大于 3000。

已知，一年有 1 月到 12 月，共 12 个不同的月份；其中 1 月、3 月、5 月、7 月、8 月、10 月、12 月有 31 个合法的日，分别为 1 日到 31 日；4 月、6 月、9 月、11 月有 30 个合法的日，分别为 1 日到 30 日。对于闰年，2 月有 29 个合法的日，分别为 1 日到 29 日；对于平年（不是闰年的年称为平年），2 月有 28 个合法的日，分别为 1 日到 28 日。

对于不大于 3000 年的年份，闰年遵循如下规律：

- 非整百年，能被 4 整除的为闰年。
- 整百年，能被 400 整除的是闰年。

要想做好这道题，你需要先回忆一下，如何使用 `scanf` 去接收一个指定了格式的字符串中的量（在这里就是 `%d` 前后的正整数）。

之后呢，你需要做很多的判断，会嵌套地用到我们已经学的 `if...else...` 的结构。你将需要：

- 判断输入日期中的月份是不是合法的月份（这里特殊情况最少，建议第一层先判断）；
- 针对每个月的日数判断输入日期中日的合法性；
- 注意抓住特殊的 2 月，在里面讨论年的问题（看是不是闰年）；

- 对于闰年，允许 29 日的存在
- 对于非闰年不能允许 29 日的存在。

样例输入

2016-2-29

样例输出

YES

样例输入：

1900-2-29

样例输出：

NO

样例输入：

2016-13-32

样例输出：

NO

提示信息

在提交给测评机测试之前，你可以自己使用“测试”功能并在弹出的框中写下你希望输入的数据，点击“运行”后等待结果。如果编译、运行顺利，你将得到结果。

如果你在测试中输入了样例输入中的文本，你还可以将自己的得到的结果与样例输出进行比对，看看自己是不是已经基本在正轨上了。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。最终尝试提交前，你最好再自己多设计几组符合题目要求的输入，例如“合法的月与不合法的日”、“合法的日与不合法的月”、“非闰年的 2 月 28 日”等等。

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。

小贴士：

- 对于 $1 < x < 3$ 这样的情况判断一定要写成 `1 < x && x < 3` 喔。
- 千万不要把 `if` 的条件写成了 `if (month == 1 || 2 || 3 || 4)` 这样的形式喔，这么写实际表达的是判断 `month` 与 `1 || 2 || 3 || 4` 的结果是否相等，而 `1 || 2 || 3 || 4` 的运算结果是“真”（也就是 1）。所以如果写成 `if (month == 1 || 2 || 3 || 4)` 实际上相当于写成了 `if (month == 1)`。