

## 勇攀高峰 / Challenge Accepted

「CACCEPT」

Kurumi 是一位资深 osu!mania 玩家。当被问起滋瓷哪种键位的时候，Kurumi 总是答道：“我可以跟你们说一句无可奉告。”

“但是你们又不高兴，我怎么办？”又一次被问到“认为 SDFJK 键位吼不吼啊”的问题之后，Kurumi 表示无奈，“问出来的问题啊，都 too simple！你们这样子是不行的！”为了摆脱这些无止境的问题，Kurumi 决定告诉新闻工作者们一些人生的经验。

Kurumi 最滋瓷的键位排列是一个五个不同字母的排列  $K_{1\dots 5}$ 。但是 Kurumi 是身经百战了，见得多了，决定让新闻工作者们猜出这个键位。对于每一次猜测，Kurumi 都只会给出非常少量的信息，所以 Kurumi 今天算是得罪了他们一下。

### 任务

对于一个未知的五个互不相同的字母排列  $K$ ，通过一系列询问确定  $K$  中五个字母的排列。

### 实现

在本题中，你不必编写一个完整的程序，而只需要实现一个过程 `newGame()` 用以进行一组  $K$  的交互并给出答案。该方法不接收任何参数。

`newGame()` 可以多次调用函数 `makeAttempt(a, b, c, d, e)` 来进行一次猜测。该函数接收五个字符类型的参数  $a, b, c, d, e$ （不必互不相同），并返回一个整数  $R = 10R_1 + R_2$  ( $0 \leq R_1, R_2 \leq 5$ )，其中：

- $R_1$  表示五个位置中有多少个位置上的字母和  $K$  中相同（不区分大小写，下同）；
- $R_2$  表示除去正确的位置上的字母，在本次猜测中还有多少个不同的字母在  $K$  中出现过。

`newGame()` 需要通过一次或多次对 `makeAttempt(a, b, c, d, e)` 的调用来确定  $K$  中五个字母的排列。当你的程序能够确定这个答案时，把  $K$  作为参数调用一次 `makeAttempt`，即 `makeAttempt(K_1, K_2, K_3, K_4, K_5)`，然后退出过程。

### 样例

考虑  $K = \text{"sdfjk"}$  的栗子。

首先 `newGame()` 被调用。它进行的工作如下：

- 调用 `makeAttempt('s', 'd', 'j', 'k', 'l')`，得到返回值 22；
- 调用 `makeAttempt('d', 's', 'j', 'k', 'l')`，得到返回值 4；
- 调用 `makeAttempt('s', 'd', 'f', 'o', 'p')`，得到返回值 30；
- 调用 `makeAttempt('s', 'd', 'f', 'k', 'p')`，得到返回值 31；
- 调用 `makeAttempt('s', 'd', 'f', 'j', 'k')`，得到返回值 50 并退出。

## 评分

评分时，本题共有 20 个预先确定的不同的  $K$ ，`newGame()` 将在同一次程序运行中被调用恰好 20 次。（因此你可以使用预处理、静态/全局变量等方式来加速程序。）

若程序发生运行时错误或运行超过时限，本次提交得分为 0；否则得分按照下面的方式计算：

$$\text{TotalScore} = \lfloor A + E \rfloor$$

其中：

- **准确度得分  $A$** ：等于在 20 组数据中获得正确答案的个数。
- **效率得分  $E$** ：
  - 若  $A \neq 20$ ：  $E = 0$ ；
  - 若  $A = 20$ ： 陪审团设定了一个猜测次数的下界  $L$ ，但  $L$  的具体数值在比赛期间不公布。假设你的程序在第  $i$  组数据上调用了  $C_i$  次 `makeAttempt` 函数，那么

$$E = 4 \cdot \sum_{i=1}^{20} \min \left( 1, \frac{L}{C_i} \right)$$

## 细节 & 本地测试

在本题的附件中包含 `grader.[c|cpp|pas]`，`caccept.[c|cpp|pas]`，`compile_[c|cpp|pas].bat` 以及一些附加文件。

其中 `grader.[c|cpp|pas]` 是用于在本地测试程序的函数库，`caccept.[c|cpp|pas]` 分别是对应语言的样例程序，实现算法时直接在其基础上进行修改即可。提交评测时只需提交该文件。修改完毕后，运行 `compile_[c|cpp|pas].bat` 来编译你的程序。编译好的程序将从文件 `caccept.in` 读取数据，调用 `newGame()`，并输出对应的运行结果（正确性和调用次数）。你可以编辑 `caccept.in` 以测试算法在不同  $K$  取值上的表现。

输入文件 `caccept.in` 不能超过 131072 行，每行包含五个字母（大小写均可）表示一组数据的  $K$ 。对于文件中的每一组数据，`newGame()` 将分别被调用一次。

## 限制

- 时间：20.0 秒（所有 20 组数据之和）
- 内存：1.0 GiB

