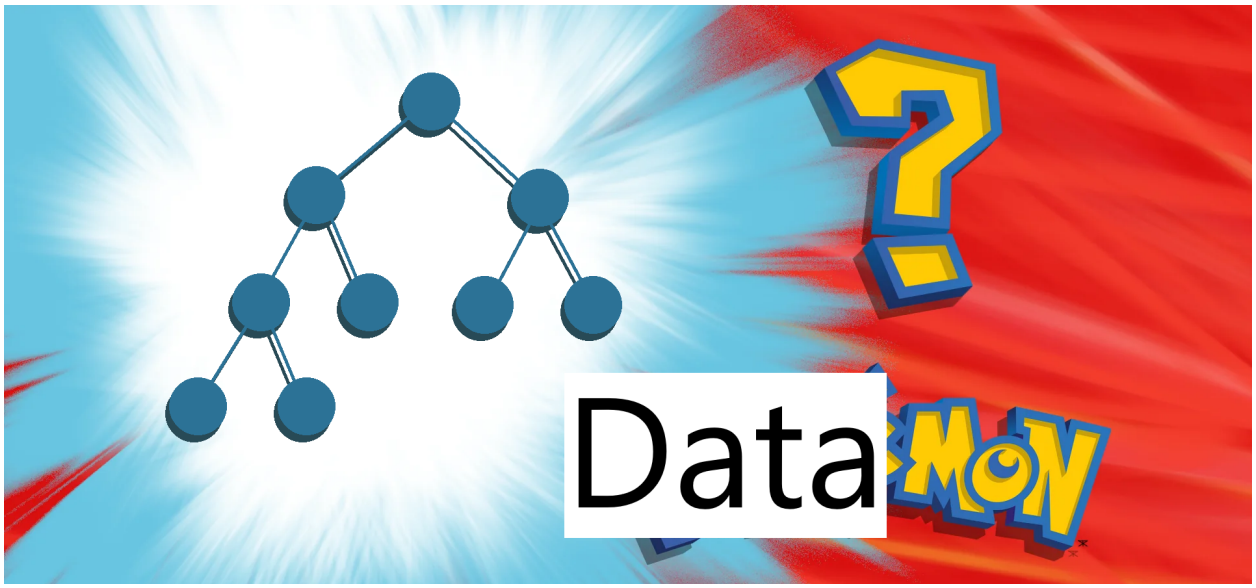
第 4 题: 这是什么数据兽?
4 分

问题标识符: `wtds`

难度等级: `1`

问题背景

你是一位勇敢的驯兽师，刚刚才抓到了自己的第一只数据兽！这并不容易，但你巧妙地躲避了动物园看守员，从而获得了自己的数据兽。然而，当你想要把这只数据兽正式登记在自己名下时，第一个问题就难倒了你：这只数据兽是什么种类?



不幸的是，自从你的手机被教授“保管”并在暗网上出售以来，你无法专心在原神里探索世界。于是你想出了完美的解决方案——上课睡大觉！因为没能专心听讲，现在你该如何辨认自己的数据兽呢?

幸好你在他们的车里发现了本教授的日记，发现他们在空闲时间热衷于研究数据兽的排便情况。希望你有限的阅读能力能帮助你从日记中收集到足够信息，辨别自己的数据兽!

问题描述

已知你拥有一只种类不明的数据兽，你想通过喂给它数字然后观察其排便情况来正确辨别它的身份。目前有三种可能的数据兽类型，它们的胃部功能如下：

- `queueon`: 始终按数据喂入的顺序排出数据，采取“ 先进先出”原则。
  （想象有一群人排队买面包，谁先来谁就先买）
- `stackeon`: 始终先排出最近喂入的数据，采取“ 先进后出”的原则。
  （想象有一叠盘子，你只会移动最上面的盘子或把盘子叠在最上面）
- `heapeon`: 始终排出其中最大的数字。
  （想象在学校中，处分最多的学生会被最先开除）

一开始，数据兽的胃部都是空的。每次互动都可以执行以下三个操作之一，可执行无数次直到辨别出数据兽的种类。

- 喂食；给数据兽喂数字：
  - 数据兽的胃部开始积累数字。
  - 评测系统会显示 `OK`。
- 排便；数据兽排出数字：
  - 不同种类的数据兽会根据其类型排出不同的数字。
  - 排出的数字从数据兽胃部移出。
  - 评测系统根据排出的数字进行判断。
  - 如果数据兽的胃部没有数字，则不再排出数字。
- 猜测数据兽的种类：
  - 如果猜测正确，评测系统会显示 `CORRECT` 并开始下一个测试用例。
  - 如果猜测错误，评测系统会显示 `WRONG_ANSWER` 并停止互动。

注：该问题以及本次活动中的所有其他问题都有 Python、Java 和 C++ 版本的模板！你可以在活动开始时提供的contest.zip压缩包中找到它们。模板将帮你处理输入输出格式，让你可以直接编写问题的解决方案！

互动格式

这是一个交互式问题！与普通问题不同，你的程序和评测系统将会同时进行。更多信息请见《活动指南》。输出时请在该帖子指导下刷新缓冲区，或运用模板代码来处理。如果在互动过程中遇到技术问题，请给我们发送请求并阐述问题！

开始时，请输入一行包含一个整数 T ，表示后面测试用例的数量。对于每一个测试用例：

A. 开始时：进行*喂养* 或*排出* 查询 。
   - 进行*喂养* 操作：
     i. 首先，按照以下格式输出一行字符串：
        `feed` i
        其 中 i 是一个整数，**0** ≤ i ≤ **100**，表示喂给数据兽的数字。
     ii. 然后读取一行，包含字符串 `OK`。
   - 进行*排便* 操作：
     i. 首先，输出一行字符串 `poop`
     ii. 然后读取一行数据兽排出的整数。

B. 结束时：进行*猜测*操作。
   - 进行 *猜测* 操作：
     i. 首先，按照以下格式输出一行字符串：
        `guess` s
        其中 s 为表示你对未知数据兽种类猜测的字符串，可以选择 `queueon`, `stackeon` 或 `heapeon`。
     ii. 然后读取一行字符串，包括 `CORRECT` 或 `WRONG_ANSWER`
        - 如果是 `CORRECT`，开始下一个测试用例。
        - 如果是 `WRONG_ANSWER`，则退出程序。

如果你的程序偏离了互动格式（例如无效查询，即当胃部为空时进行*排出* 查询等），评测系统将会判定为 `WRONG_ANSWER`。一旦程序读取到 `WRONG_ANSWER` ，你应退出程序并接收错答案错误的判定。

数据范围

T = 100
在所有的喂养操作中，喂入的数字 i 必须是一个满足0 ≤ i ≤ 100的整数。
数据兽只可能属于 `queueon`, `stackeon` 或 `heapeon`三个种类。

互动示例

这里的行距只是为了强调互动的顺序，请勿在互动的每行之间输出空白行。

样例输入	样例输出
100	
	feed 5
OK	
	feed 7
OK	
	poop
5	
	feed 6
OK	
	poop
7	
	guess queueon
CORRECT	
	feed 5
OK	
	feed 5
OK	
	feed 5
OK	
	poop
5	
	poop
5	
	guess heapeon
CORRECT	
	feed 9
OK	
	feed 7
OK	
	feed 15
OK	
	poop
15	
	guess stackeon
WRONG_ANSWER	

样例解释

首先输出测试用例的总数100，随后开始逐一进行判定。

在测试用例 #1 中，程序首先进行了一次喂食操作，将数字 5 喂给数据兽，评测系统回应 `OK` 。然后程序又进行了一次喂食操作，这次喂入的数字是7。之后进行了一次排便操作，数据兽排出了数字 5。接着，程序再次喂入数字 6，然后数据兽排出了数字 7。最后，程序猜测这只数据兽属于 `queueon` 种类，评测系统确认了这一猜测，显示 `CORRECT` 。

测试用例 #2 中，程序对数据兽连续三次喂入数字 5。在此期间，程序进行了两次排便操作，每次都排出了数字 5。随后，程序猜测数据兽属于 `heapeon` 种类，评测系统同样显示 `CORRECT`。尽管程序正确猜测了 `heapeon` 类型，但在这一系列操作中，`stackeon` 或 `queueon` 类型的数据兽也可能有同样的表现。

测试用例 #3 中，程序首先进行了三次喂食操作，分别喂给数据兽数字 9、7 和 15。随后程序进行了排便操作，排出了数字15。最后，程序猜测数据兽是一种 `stackeon`，但评测系统判定为 `WRONG_ANSWER`，并退出程序。虽然观察到的表现与 `stackeon` 一致，但 `heapeon`也能产生这种表现，而 `heapeon` 正好是该数据兽的种类。

尽管有 100 个测试用例，但互动在三个测试用例之后就会停止，因为测评系统接收到了 `WRONG_ANSWER` 的判定。

该页留白。