



天平

Amina有6枚重量互不相同的硬币（1到6进行编号）。为了将这些硬币按重量排序，Amina设计了一种新的天平。

传统的天平有两个秤盘，使用时在每个秤盘上放一枚硬币，即可称量出哪一枚硬币比较重。

Amina的新天平比较复杂，它有4个秤盘，分别记为A、B、C、D。这个新天平有4种用法，每种用法回答一个关于硬币重量的问题。使用这个新天平时，秤盘A、B、C上必须各放一枚硬币。注意：当使用新天平的第4种用法时，秤盘D上也必须放入一枚硬币。

新天平的4种用法分别回答下面的四个问题：

1. 秤盘A、B、C上的硬币，哪个最重？
2. 秤盘A、B、C上的硬币，哪个最轻？
3. 秤盘A、B、C上的硬币，哪个的重量居中（既不是最重的，也不是最轻的）？
4. 如果秤盘A、B、C上存在比秤盘D上重的硬币，比秤盘D上重的硬币中哪个最轻？如果A、B、C上不存在比D上重的硬币，那么秤盘A、B、C上的硬币，哪个最轻？

任务

编写一个程序按硬币的重量对其排序，程序中可以使用Amina的新天平来比较硬币的重量。该程序需要解决若干个测试用例，每个测试用例对应一组新的6个硬币的组合。

你的程序需要实现函数init和orderCoins。每次运行你的程序，评测程序首先调用一次init函数（只调用一次），该函数告诉你评测用例的数目，在此函数中你可以初始化任意变量。接着，评测程序针对每一个评测用例调用一次orderCoins()函数。

- init(T)
 - T：在这次运行中，你的程序需要求解的测试用例的数目。T是一个整数，取值范围是1, ..., 18。
 - 该函数没有返回值。
- orderCoins()
 - 对于每一个测试用例，该函数会被调用一次且仅被调用一次。
 - 该函数通过调用函数getHeaviest()、getLightest()、getMedian()、getNextLightest()来确定硬币的正确排序
 - 一旦知道了硬币的正确排序，该函数即可调用函数answer()。
 - 调用函数answer()后，本函数orderCoins()应该立即返回，且无返回值。

你的程序中可以使用下列函数（grader functions）：

- answer(w) — 调用此函数给出你找到的硬币排序。

- w : 长度为6的数组，包含了硬币的正确排序，即 $w[0]$ 到 $w[5]$ 中是按照硬币重量从小到大的顺序存放硬币的编号（硬币的编号是从1到6的）。
- 对于每个测试用例，你的程序只能在函数`orderCoins()`中调一次`answer(w)`函数。
- 该函数没有返回值。
- `getHeaviest(A, B, C)`, `getLightest(A, B, C)`, `getMedian(A, B, C)` — 这三个函数分别对应Amina的天平的第1、2、3种用法。
 - A, B, C : 分别表示放在秤盘 A, B, C 上的硬币的编号。 A, B, C 是3个互不相同的整数，每个整数的取值范围都是1到6。
 - 每个函数均返回数字 A, B, C 中的一个，表示符合条件的硬币的编号。例如，`getHeaviest(A, B, C)`返回3个硬币中最重的那个硬币的编号。
- `getNextLightest(A, B, C, D)` — 该函数对应Amina的天平的第4种用法。
 - A, B, C, D : 分别表示放在秤盘 A, B, C, D 上的硬币的编号。 A, B, C, D 是4个互不相同的整数，取值范围是1到6。
 - 该函数返回 A, B, C 中的一个数字: 表示Amina的天平第4种用法选出的硬币的编号，即返回的硬币编号是秤盘 A, B, C 上比 D 上硬币重的硬币中最轻的那个硬币的编号，或者，如果 A, B, C 上的硬币都轻于 D 上的硬币，那么返回的是秤盘 A, B, C 中最轻的那个硬币的编号。

评分标准

本题没有子任务。你的得分与你称量的次数（调用函数`getLightest()`、`getHeaviest()`、`getMedian()`、`getNextLightest()`的总次数）有关。

每次运行时，你的程序会针对多个测试用例运行多次。假设 r 是你的程序运行的次数，它是由测试数据决定的。如果你的程序在任何一次运行中对任意一个测试用例给出的硬币排序结果不正确，那么你将得到0分，否则，按照下面的规则进行评分。

假设 Q 是使用Amina的天平对任意的6个硬币进行排序所需要称量的最小次数。为了让本题更具挑战性，这里不告知 Q 的值。

假设所有运行中所有测试用例中的最大称量次数为 $Q + y$ ， y 是整数。然后，考虑你程序的一次运行，设这次运行中所有 T 个测试用例中最大的称量次数为 $Q + x$ ， x 是非负整数。（对每个测试用例，如果你的称量次数比 Q 少，那么 $x = 0$ 。）那么，这次运行你的得分是 $\frac{100}{r((x+y)/5+1)}$ ，向下取整保留到小数点后两位数字。

特别的，如果你的程序在每次运行中对任意测试用例都最多称量 Q 次，你将得到100分。

.....

.....

.....

.....

样例

假设硬币重量从小到大的顺序是**3 4 6 2 1 5**。

函数调用	返回值	说明
getMedian(4, 5, 6)	6	6号硬币的重量在4、5、6号硬币中居中。
getHeaviest(3, 1, 2)	1	1号硬币是1、2、3号硬币中最重的。
getNextLightest(2, 3, 4, 5)	3	2、3、4号硬币都比5号硬币轻，所以返回2、3、4号硬币中最轻的，即3号。
getNextLightest(1, 6, 3, 4)	6	1、6号硬币均比4号硬币重，返回它们两个中最轻的那个，即返回6。
getHeaviest(3, 5, 6)	5	5号硬币是3、5、6号硬币中最重的。
getMedian(1, 5, 6)	1	1号硬币的重量在1、5、6号硬币中居中。
getMedian(2, 4, 6)	6	6号硬币的重量在2、4、6中居中。
answer([3, 4, 6, 2, 1, 5])		程序找到的正确的排序结果。

Sample grader

sample grader以下列格式读入数据：

- 第1行： T — 测试用例的数目
- 第2行到第 $T + 1$ 行：每行6个互不相同的整数，分别表示6个硬币的编号，按照重量从小到大的顺序排列

例如，包含2个测试用例（硬币按重量从小到大排序分别是**1 2 3 4 5 6**和**3 4 6 2 1 5**）输入数据格式如下：

```
2
1 2 3 4 5 6
3 4 6 2 1 5
```

sample grader输出作为answer()函数参数的数组。