天平 scales

Amina 有 6 個硬幣, 編號從 1 到 6, 這些硬幣的重量互不相同。Amina 想按照重量對硬幣進行排序。為此, 他發明了一種新的天平。

傳統的天平有兩個秤盤, 使用時在每個秤盤上放一枚硬幣, 天平可以量出那一枚硬幣比較重。

Amina 的新天平比較複雜, 它有 4 個秤盤, 分別標記為 A、B、C、D。這個新天平有 4 種用法, 每種用法可以回答一個關於硬幣的不同問題。使用新天平時, 秤盤 A、B、C 上必須各放一枚硬幣。特別的, 當使用新天平的第 4 種用法時, 秤盤 D 上也必須放入一枚硬幣。

新天平的 4 種用法分別回答下面的四個問題：

1. 秤盤 A、B、C 上的硬幣, 那個最重？
2. 秤盤 A、B、C 上的硬幣, 那個最輕？
3. 秤盤 A、B、C 上的硬幣, 那個的重量居中？（即不是最重的, 也不是最輕的。）
4. 考慮秤盤 A、B、C 上比秤盤 D 上重的那些硬幣, 如果存在, 其中那個是最輕的？ 如果不存在, 秤盤 A、B、C 中的硬幣, 那個是最輕的？

任務

編寫一個程式根據硬幣的重量對其排序, 該程式可以使用 Amina 的天平來比較硬幣的重量。你的程式需要解決若干個測試用例, 每個測試用例對應一組新的6個硬幣的組合。

你的程式需要實現函數 `init` 和 `orderCoins`。 每次運行你的程式, 評測程式首先調用一次 `init` 函數（只調用一次）, 通過該函數告訴你評測用例的數目, 在此函數中你可以初始化任意變數。 接著, 評測程式針對每一個評測用例調用一次 `orderCoins()` 函數。

- `init(T)`
  - `T`：在這次運行中, 你的程式需要解決的測試用例的數目。`T` 是一個整數, 其範圍是 $1, \ldots, 18$。
  - 該函數沒有返回數
- `orderCoins()`
  - 對於每一個測試用例, 該函數會被調用一次且僅一次。
  - 該函數通過調用函數 `getHeaviest()`、`getLightest()`、`getMedian()`、`getNextLightest()` 來確定 Amina 的硬幣的正確順序
  - 該函數一旦知道了硬幣的正確順序, 即可調用函數 `answer()` 給出正確的順序。
  - 調用函數 `answer()` 後, 本函數 `orderCoins()` 應該返回, 且本函數無返回數。

你的程式中可以調用評測程式給你的下列函數：

- `answer(W)` — 你的程式調用此函數給出你找到的硬幣順序。
  - `W`: 長度為 6 的陣列, 該陣列包含硬幣的正確順序, 即 `W[0]` 到 `W[5]` 中是按照硬幣重量從小到大存放的硬幣的編號（硬幣的編號是從 1 到 6 的)。
  - 對於每個測試數據, 你的程式只能在函數 `orderCoins()` 中調這函數一次。
  - 該函數沒有返回數。
- `getHeaviest(A, B, C)`, `getLightest(A, B, C)`, `getMedian(A, B, C)` — 這 3 個函數分別對應 Amina 的天平的第 1、2、3 種用法。
  - `A`, `B`, `C`: 分別表示放在秤盤 A、B、C 上的硬幣的編號。`A`、`B`、`C` 是 3 個互不相同的整數, 每個數的範圍都是 1 到 6。
  - 每個函數返回數字 `A`、 `B`、 `C` 中的一個, 表示符合條件的硬幣的編號。 例如, `getHeaviest(A, B, C)` 返回 3 個硬幣中最重的那個硬幣的編號。
- `getNextLightest(A, B, C, D)` — 該函數對應 Amina 的天平的第4種用法。
  - `A`, `B`, `C`, `D`: 分別表示放在秤盤 A、B、C、D 上的硬幣的編號。`A`、`B`、`C`、`D` 是 4 個互不相同的整數, 其範圍是 1 到 6。
  - 該函數返回 `A`、`B`、`C` 中的一個數字: 表示 Amina 的天平第 4 種用法選出的硬幣的編號, 即返回的硬幣編號是秤盤 A 、B、 C 上比 D 秤盤上硬幣重的硬幣中最輕的那個硬幣編號, 或者, 如果 A、B、C 上的硬幣都輕於 D 上的硬幣, 那麼返回的是秤盤 A、B、C 中最輕的那個硬幣的編號。

評分標準

本題沒有子任務。你的得分與你稱量了多少次（調用函數 `getLightest()`、`getHeaviest()`、 `getMedian()`、 `getNextLightest()` 的總次數）有關。

每次運行時, 你的程式會針對多個測試用例運行多次。 假設 r 是你的程式運行的次數, 它是由測試資料決定的。 如果你的程式在任何一次運行中對任意一個測試用例給出的硬幣排序結果不正確, 那麼你將會得到 0 分, 否則, 按照下面的規則進行評分。

假設 Q 是使用 Amina 的天平對任意的 6 個硬幣進行排序所需要稱量的最小次數。 為了讓本題更具挑戰性, 這裡不告知 Q 的內容。

假設所有運行中所有測試用例中的最大稱量次數為 $Q + y$, y 是整數。然後, 考慮你程式的一次運行, 設這次運行中所有 T 個測試用例中最大的稱量次數為 $Q + x$, x 是非負整數。（對每個測試用例, 如果你的稱量次數比 Q 少, 那麼 $x = 0$。）那麼, 這次運行你的得分是 $\frac{100}{r((x+y)/5+1)}$, 向下取整保留小數點後兩位元數字。

特別的, 如果你的程式在每次運行中對任意測試用例都最多稱量 Q 次, 你將得到 100 分。

樣例

假設硬幣重量從小到大的順序是 $3\ 4\ 6\ 2\ 1\ 5$。

函式呼叫	返回數	說明
getMedian(4, 5, 6)	6	6 號硬幣的重量在 4、5、6 號硬幣中居中。
getHeaviest(3, 1, 2)	1	1 號硬幣是 1、2、3 號硬幣中最重的。
getNextLightest(2, 3, 4, 5)	3	2、3、4 號硬幣都比5號硬幣輕, 所以返回 2、3、4 號硬幣中最輕的, 即 3 號。
getNextLightest(1, 6, 3, 4)	6	1、6 號硬幣均比 4 號硬幣重, 返回它們兩個中最輕的那個, 即返回 6。
getHeaviest(3, 5, 6)	5	5 號硬幣是 3、5、6 號硬幣重最重的。
getMedian(1, 5, 6)	1	1 號硬幣的重量在 1、5、6 號硬幣中居中。
getMedian(2, 4, 6)	6	6 號硬幣的重量在 2、4、6 中居中。
answer([3, 4, 6, 2, 1, 5])		程式找到的正確的排序結果。

Sample grader

sample grader 以下列格式讀入資料：

- 第 1 行：T —- 測試用例的數目
- 第 2 行到第 $T+1$ 行：每行 6 個互不相同的整數, 分別表示 6 個硬幣的編號, 按照重量從小到大的順序排列

例如, 包含2個測試用例（硬幣的排序分別是的 $1\ 2\ 3\ 4\ 5\ 6$ 和 $3\ 4\ 6\ 2\ 1\ 5$）輸入資料格式如下：

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

sample grader 輸出作為 `answer()` 函數參數的陣列。