

秤重

Amina有六個硬幣，編號1到6。他知道硬幣的重量都互不相同。他想要將這些硬幣依據重量排序。因此，他開發了一種新式的天平秤。

傳統的天平有兩個置物盤，使用時在每個置物盤各放一個硬幣，天平就能判斷那個硬幣較重。

Amina的新式天平秤較為複雜，它有四個置物盤，標示為 A, B, C, D 。新式天平秤有四個不同的設定，每個設定可以回答一個關於硬幣的問題。使用新式天平時，Amina必須在置物盤 A, B, C 各放置正好一個硬幣。此外，在使用第四個設定時，他也必須在置物盤 D 放置正好一個硬幣。

透過這四個設定，新式天平將回答下列四個問題：

1. 置物盤 A, B, C 中的硬幣那個最重？
2. 置物盤 A, B, C 中的硬幣那個最輕？
3. 置物盤 A, B, C 中的硬幣那個重量介於中間？(這個硬幣既非三者之中最重的，也非最輕的)
4. 在置物盤 A, B, C 的硬幣中，只考慮比置物盤 D 的硬幣重的，如果有符合此條件的硬幣，那一個是其中最輕的？反之，如果沒有符合此條件的硬幣，那麼置物盤 A, B, C 中的硬幣那個是最輕的？

Task

寫程式來將Amina的六個硬幣依據重量排序。程式可以詢問Amina的新式天平來比較硬幣的輕重。你的程式將會收到好幾筆測試資料，每一筆測資對應到一組全新的六個硬幣。

你的程式必須實作 `init` 和 `orderCoins` 兩個函數。你的程式在每一次執行期間，`grader` 評分程式首先將會呼叫 `init` 正好一次，這讓你知道測資有幾筆，並且讓你可以設定變數的起始內容。然後`grader` 評分程式會在每筆測資中呼叫 `orderCoins()` 一次。

- `init(T)`
 - T : 你的程式在這次執行期間要解決的測資筆數。 T 是 $1, \dots, 18$ 範圍中的整數。
 - 這個函數沒有回傳。
- `orderCoins()`
 - 這個函數在每筆測資中只會被呼叫正好一次。
 - 這個函數必須透過呼叫`grader`函數 `getLightest()`, `getHeaviest()`, `getMedian()`, and/or `getNextLightest()` 來決定Amina硬幣的正確排序。
 - 這個函數一旦知道了硬幣的正確排序，就必須呼叫`grader`函數 `answer()` 來回報結

果。

- 在呼叫 `answer()` 之後，`orderCoins()` 函數必須返回。它沒有回傳。

你可以在你的程式中，呼叫下列的 `grader` 函數：

- `answer(W)` — 你的程式必須用這個函數回報所發現的答案。
 - `W`: 長度6的陣列，包含硬幣的正確順序。`W[0]` 到 `W[5]` 是硬幣的編號(也就是1到6的數字)，由最輕排到最重。
 - 你的程式只能從 `orderCoins()` 中呼叫這個函數，每筆測資呼叫一次。
 - 這個函數沒有回傳。
- `getHeaviest(A, B, C)`, `getLightest(A, B, C)`, `getMedian(A, B, C)` — 這些對應到Amina新式天平的第1, 2, 3個設定。
 - `A, B, C`: 放在置物盤 **A, B, C** 中的硬幣。`A, B, C` 必須是三個互不相同的整數，且介於1到6之間。
 - 每個函數回傳 `A, B, C` 之一的編號: 也就是符合條件的硬幣編號。例如 `getHeaviest(A, B, C)` 回傳三個硬幣中最重一個的編號。
- `getNextLightest(A, B, C, D)` — 這對應到Amina新式天平的第4個設定。
 - `A, B, C, D`: 放在置物盤 **A, B, C, D** 中的硬幣。`A, B, C, D` 必須是四個互不相同的整數，且介於1到6之間。
 - 這個函數回傳 `A, B, C` 之一的編號: 這是新式天平依據上述第4個設定，所選出的硬幣編號。也就是說，回傳的硬幣是置物盤 **A, B, C** 的硬幣裏，比置物盤 **D** 的硬幣重的那幾個中最輕的一個。或者是，如果沒有一個比置物盤 **D** 的硬幣重，回傳的硬幣就純粹是置物盤 **A, B, C** 的三個硬幣中最輕的一個。

Scoring

這一題裏沒有 `subtasks`。你的分數將會根據你的程式稱重的次數(呼叫 `grader` 函數 `getLightest()`, `getHeaviest()`, `getMedian()` and/or `getNextLightest()` 的合計次數)。

你的程式會被執行許多次，每次執行會用到許多筆測試資料。假設 r 是你的程式執行的次數，測資中這個次數是固定的。如果你的程式在任何一次執行中的任何一筆測資沒有將硬幣正確排序，則得分為0，否則每次執行結果依據下列方式評分。

假設 Q 是: 用Amina新式天平稱重 Q 次後，就能將任何六個硬幣排序完成的最小數字。為了讓任務更具挑戰性，我們在這裡不透漏 Q 是多少。

假設在所有執行的所有測資中，稱重的最大次數是 $Q + y$ ， y 是某個整數。然後考慮你的程式的某一次執行，讓這次執行的全部 T 筆測資中，稱重的最大次數是 $Q + x$ ， x 是某個非負數的整數(如果你每個測資都稱重不到 Q 次，則 $x = 0$)，則這次執行的得分是

$$\frac{100}{r((x+y)/5+1)}$$
 無條件捨去計算至小數點後兩位。

特別是，如果你的程式每次執行的每個測資中，都稱重最多 Q 次，你將獲得100分。

Example

假設硬幣從最輕到最重的排序是 **3 4 6 2 1 5**

Function call	Returns	Explanation
getMedian(4, 5, 6)	6	硬幣 6 是硬幣 4, 5, 6 的中間者
getHeaviest(3, 1, 2)	1	硬幣 1 是硬幣 1, 2, 3 中最重者
getNextLightest(2, 3, 4, 5)	3	硬幣 2, 3, 4 都比硬幣 5 輕，所以回傳當中最輕的 (3)
getNextLightest(1, 6, 3, 4)	6	硬幣 1 和 6 比硬幣 4 重，在硬幣 1 和 6 之間, 硬幣 6 是最輕的一個。
getHeaviest(3, 5, 6)	5	硬幣 5 是硬幣 3, 5 和 6 中最重者
getMedian(1, 5, 6)	1	硬幣 1 是硬幣 1, 5 和 6 的中間者
getMedian(2, 4, 6)	6	硬幣 6 是硬幣 2, 4 和 6 的中間者
answer([3, 4, 6, 2, 1, 5])		程式找到了這筆測資的正確解答

Sample grader

範例評分程式依下列格式讀取輸入資料:

- 第 1 行: T — 測資的筆數
- 第 2 至 $T + 1$ 行的每一行: 一連串 6 個 1 到 6 的不同數字: 硬幣從最輕到最重的順序。

例如，一個含有兩筆測資的輸入資料，裡面硬幣的順序為 **1 2 3 4 5 6** 和 **3 4 6 2 1 5** 看起來如下所列:

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

範例評分程式印出傳遞給 `answer()` 函數做為參數的陣列